

Diplomarbeit

# **Echtzeitsimulation detaillierter Fahr- und Antriebsstrangdynamik**

Dirk Mauermann

08. Juli 2004

# Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als Diplomand in der Forschung am Institut für Robotik und Mechatronik des Deutschen Zentrums für Luft- und Raumfahrt e. V. (DLR) in Oberpfaffenhofen. Die Betreuung von Seiten der HTWK Leipzig übernahm dankenswerter Weise das Institut für Mess-, Steuerungs- und Regelungstechnik des Fachbereichs Elektrotechnik und Informationstechnik.

Herrn Prof. Dr.-Ing. habil. K.-P. Schulze möchte ich für die langjährige Unterstützung und Förderung während meines Studiums danken. Insbesondere danke ich für die Motivation und engagierte Betreuung zu dieser Arbeit.

Mein besonderer Dank gilt Herrn Dipl.-Ing. Christian Schweiger, der mir während der gesamten Dauer der Arbeit mit kompetenter Hilfe zur Seite stand. Allen Kolleginnen und Kollegen vom DLR danke ich für die konstruktive Kritik und gute Zusammenarbeit.

Abschließend möchte ich mich in besonderer Weise bei meiner Lebenspartnerin Katrin sowie meinen Eltern bedanken, die immer hinter mir stehen und ohne die diese Arbeit nicht entstanden wäre.

Leipzig, im Juli 2004

Dirk Mauermann

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Problemstellung</b>	<b>3</b>
2.1	Fahrdynamik . . . . .	3
2.2	Antriebsstrangdynamik . . . . .	4
2.3	Fahrzeug – Fahrer – Umwelt . . . . .	5
2.4	Regelungsentwicklung . . . . .	6
2.5	Simulationstechnik . . . . .	7
<b>3</b>	<b>Stand der Technik mit Ausblick</b>	<b>9</b>
3.1	Fahrdynamikregelungen . . . . .	9
3.1.1	Antiblockiersysteme . . . . .	10
3.1.2	Antriebsschlupfregelung . . . . .	11
3.1.3	Elektronische Bremskraftverteilung und Bremsassistent . . . . .	12
3.1.4	Fahrdynamikregelung . . . . .	13
3.1.5	Aktive Fahrwerke . . . . .	15
3.1.6	Adaptive Cruise Control (ACC) . . . . .	15
3.1.7	ESP II – Fahrdynamik der nächsten Generation . . . . .	16
3.2	Antriebsstrangdynamik . . . . .	19
3.2.1	Verbrennungsmotoren . . . . .	19
3.2.2	Automatikgetriebe . . . . .	21
3.2.3	Antriebsstrangsteuerungen . . . . .	22
3.3	Die Zukunft des Automobils . . . . .	24
3.3.1	Mechatronik . . . . .	25
3.3.2	Aktive Chassis-Systeme . . . . .	26
3.3.3	Global Chassis Control . . . . .	28
3.3.4	Das X-By-Wire-Projekt . . . . .	30
3.3.5	Ausblick . . . . .	31

3.4	Elektronik im Kraftfahrzeug . . . . .	31
3.5	Simulation in der Automobilentwicklung . . . . .	33
3.5.1	Entwicklungsprozess . . . . .	34
3.5.2	Simulations-Methoden . . . . .	37
3.5.3	Virtuelle Produktentwicklung . . . . .	39
3.5.4	Rapid Prototyping . . . . .	39
3.5.5	Simulationswerkzeuge . . . . .	40
3.6	Zusammenfassung . . . . .	42
<b>4</b>	<b>Aufgabenstellung</b>	<b>43</b>
<b>5</b>	<b>Objektorientierte Modellierung</b>	<b>45</b>
5.1	Einführung . . . . .	45
5.2	Modellierung mit Dymola und Modelica . . . . .	46
5.3	Vergleich mit Matlab/Simulink . . . . .	50
<b>6</b>	<b>Antriebsstrangmodell</b>	<b>52</b>
6.1	Motor . . . . .	52
6.1.1	Drehträgheit . . . . .	53
6.1.2	Motor-Drehmomentmodell . . . . .	54
6.1.3	Motorsteuerung . . . . .	55
6.1.4	Anlasser . . . . .	56
6.1.5	Bus . . . . .	56
6.2	Automatikgetriebe . . . . .	57
6.2.1	Drehmomentwandler . . . . .	58
6.2.2	Elektronische Getriebesteuerung (EGS) . . . . .	59
6.2.3	Lepelletier-Radsatz . . . . .	61
6.3	Antriebsstrang mit 3D-Effekten . . . . .	64
6.3.1	Verbrennungsmotor . . . . .	65
6.3.2	Lepelletier-Radsatz . . . . .	66
<b>7</b>	<b>Fahrdynamikmodell</b>	<b>67</b>
7.1	Koordinatensysteme . . . . .	67
7.2	Struktur des Fahrdynamikmodells . . . . .	68
7.2.1	Chassis . . . . .	69
7.2.2	Fahrwerk mit Lenkung . . . . .	71
7.2.3	Reifen . . . . .	73

7.3	Abtrieb . . . . .	74
7.4	Umwelt . . . . .	76
7.5	Geschwindigkeitsmessung . . . . .	76
7.6	Fahrdynamikmodell mit 3D-Effekten . . . . .	77
<b>8</b>	<b>Fahrermodell</b>	<b>78</b>
8.1	Longitudinal-Fahrer . . . . .	78
8.2	Lateral-Fahrer . . . . .	79
<b>9</b>	<b>Detailliertes Fahrzeug-Echtzeitmodell</b>	<b>81</b>
9.1	Gesamtfahrzeugmodell . . . . .	81
9.2	Parametrisierung . . . . .	82
9.3	Initialisierung . . . . .	85
<b>10</b>	<b>Simulationsergebnisse</b>	<b>88</b>
10.1	Verifizierung des Fahrzeugmodells . . . . .	88
10.1.1	Simulationsdurchführung mit <i>Dymola</i> . . . . .	88
10.1.2	Integrationsverfahren und Eventbehandlung . . . . .	89
10.1.3	Ausgewählte Simulationsergebnisse . . . . .	90
10.2	Echtzeitsimulation . . . . .	95
10.2.1	Entwicklungsumgebung . . . . .	95
10.2.2	Anpassung der Simulationseinstellungen . . . . .	96
10.2.3	Ausgewählte Simulationsergebnisse . . . . .	97
10.3	Fahrzeugmodell mit 3D-Effekten im Antriebsstrang . . . . .	102
10.3.1	Vergleich der Simulationsergebnisse . . . . .	104
<b>11</b>	<b>Wesentliche Ergebnisse und Schlussfolgerungen</b>	<b>107</b>
<b>12</b>	<b>Zusammenfassung und Ausblick</b>	<b>110</b>
<b>A</b>	<b>Matlab-Skript</b>	<b>112</b>
<b>B</b>	<b>Dymola-Skript</b>	<b>113</b>

# 1 Einleitung

Seit Anfang der 70er Jahre ist die Entwicklung von Kraftfahrzeugen geprägt von einem rasanten Anstieg des Einsatzes von Elektronik und Software. Dieser Trend wird sich, getrieben von steigenden Kunden- und Umweltaforderungen, weiter fortsetzen. Im Vordergrund stehen bei der Entwicklung neuer Kraftfahrzeuge die Verringerung von Kraftstoffverbrauch und Schadstoffemission, sowie die Erhöhung von Fahrsicherheit, Agilität und Fahrkomfort. Des Weiteren zeichnet sich aufgrund der technischen Möglichkeiten und des starken Wettbewerbs zwischen den Automobilherstellern um Marktanteile eine Tendenz zu immer kürzeren Entwicklungszeiten und größerer Modellvielfalt ab [Hol01]. Zur Erreichung dieser Herausforderungen und aufgrund der rasanten Entwicklung auf dem Gebiet der Rechensysteme werden im Automobilentwicklungsprozess vermehrt Software-Werkzeuge, so genannte *Computer-Aided-Engineering*-Werkzeuge (*CAE*-Werkzeuge), eingesetzt.

Durch gezielte Entwicklungen in den einzelnen Teilbereichen wurde in den letzten Jahren ein hoher Stand der Technik in der Automobilindustrie erreicht. Aufgrund dieser Erfolge in der Vergangenheit ist ein spürbarer Innovationsdruck vorgegeben [WN03]. Die Entwicklung auf den einzelnen Teilgebieten ist jedoch so weit fortgeschritten, dass durch isolierte Maßnahmen kaum weitere spürbare und wirtschaftliche Fortschritte zu erzielen sind. Um bei der weiteren Entwicklung des dynamischen Systems „Fahrzeug“ langfristig erfolgreich zu sein, müssen im Entwicklungsprozess die Systemgrenzen zunehmend über das eigentliche Problem hinaus ausgedehnt werden. Es muss das gesamte System, bestehend aus Fahrer, Fahrzeug und Umgebung, fachdisziplinübergreifend betrachtet werden.

Durch die dynamische Entwicklung der Elektronik wurden viele mechanische und hydraulische Funktionen im Automobil in den letzten Jahren durch Software-Funktionen ersetzt, wie z. B. die Automatikgetriebesteuerung. Heutzutage werden inzwischen nahezu alle Funktionen des Fahrzeugs elektronisch gesteuert, geregelt und/oder überwacht. Historisch bedingt realisierten die Automobilhersteller neue Funktionen in separaten elektronischen Steuergeräten<sup>1</sup>. Die Realisierung von Funktionen durch Software bietet beim Entwurf einzigartige Freiheitsgrade. Neue Funktionen können aber nur effizient durch weiträumige Vernetzung der Steu-

---

<sup>1</sup>Im Automobilbereich wird unabhängig von einer realisierten Steuerung oder Regelung von einem Steuergerät gesprochen.

ergeräte entwickelt werden. Zusätzlich lassen sich dadurch die Herstellungskosten verringern. Im heutigen Entwicklungsprozess werden die entsprechenden Steuerungs- und Regelungsfunktionen eigenständig entwickelt und erst am Ende in das Gesamtfahrzeug integriert und darin getestet. Fehler durch die Überlagerung von Regelungen können so erst sehr spät entdeckt werden. Es bestehen daher Bemühungen, schon frühzeitig komplex vernetzte Regelungen, wie z. B. die *Fahrdynamikregelung (ESP)*, durch geeignete Simulationen zu untersuchen. Des Weiteren sind für Komfort- und Sicherheitsbetrachtungen dynamische Effekte vom Antriebsstrang, die sich auf die Fahrdynamik auswirken, von großem Interesse.

Für die Simulation von Computermodellen ergibt sich in der Automobilentwicklung ein sehr vielfältiger Einsatzbereich. Dabei müssen die Freiheitsgrade bei der Simulation unter genauer Beachtung der Aufgabenstellung hinsichtlich Kosten und Genauigkeit gelöst werden. Die Komplexität der eingesetzten Modelle und der damit verbundene Rechenaufwand bestimmen in Kombination mit der verwendeten Rechnerhardware die Rechengeschwindigkeit des Modells. Bei Echtzeitsimulationen ist daher besonders bei der Modellierung des Systems auf einen überlegten Kompromiss zwischen Rechenzeit und Modellgüte zu achten. Die Echtzeitsimulation mit objektorientierten physikalischen Modellen gewinnt daher zum Test und zur Optimierung realer Systeme an Bedeutung. Mittels einer Hardware-in-the-Loop-Simulation können so gefahrlos, aufwandsarm und reproduzierbar Steuergeräte getestet werden.

In der vorliegenden Arbeit wird ein Gesamtfahrzeugmodell betrachtet und auf Echtzeitfähigkeit hin untersucht. Dazu wird im Kapitel 2 die Problemstellung für diese Arbeit aufgezeigt. Im Kapitel 3 wird ein Überblick über den Stand der Technik der Steuerungs- und Regelungssysteme und der Simulationstechnik für den Automobilbereich gegeben. Die verschiedenen Steuerungen und Regelungen für den Antriebsstrang und die Fahrdynamik werden dabei ausführlich beschrieben. Mit der genauen Aufgabenstellung im Kapitel 4 wird der Hauptteil der Arbeit eingeleitet. Im Kapitel 5 werden ausführlich das Simulationssystem *Dymola* und die Modellierungssprache *Modelica* erläutert. In den Kapiteln 6, 7 und 8 werden die Modellkomponenten des Fahrzeugmodells beschrieben. Das Gesamtfahrzeugmodell und deren Parametrisierung und Initialisierung mit *Dymola* werden im Kapitel 9 dargestellt. Danach erfolgt, im Kapitel 10, die Darstellung ausgewählter Simulationsergebnisse und eine Untersuchung der dreidimensionalen Momente im Antriebsstrang auf die Fahrdynamik. Im Kapitel 11 werden die wesentlichen Ergebnisse aus der Simulation hervorgehoben. Zum Abschluss der Arbeit wird im Kapitel 12 eine Zusammenfassung mit Ausblick gegeben.

## 2 Problemstellung

Die Entwicklung eines detaillierten Gesamtfahrzeugmodells unter Beachtung der Echtzeitfähigkeit stellt eine große Herausforderung dar. Im Folgenden werden die Probleme bei der Modellierung und Simulation aufgezeigt.

### 2.1 Fahrdynamik

Die Verbesserung der Fahrsicherheit und Fahrstabilität bei gleichzeitig verbessertem Fahrkomfort stellt eines der wesentlichen Ziele der Entwicklung von Kraftfahrzeugen dar. Der Einsatz von Regelungssystemen zur Verringerung von mechanischen und akustischen Schwingungen sowie zur Stabilisierung des Fahrzeugs in kritischen Situationen soll sowohl die Fahrsicherheit als auch den Fahrkomfort steigern. Zur Erreichung dieser Ziele benötigt man Modelle, die Bewegungsgleichungen in Längs-, Quer- und Vertikalrichtung berücksichtigen. Besonders sind die dynamischen Brems- und Seitenführungskräfte zwischen Reifen und Fahrbahn zu betrachten. Diese Modelle werden mit rechnergestützten Entwicklungswerkzeugen (*CAE*) entwickelt und simuliert. Zur Modellierung der Systeme sind abstrakte physikalisch-mathematische Modelle notwendig. Die Fahrdynamik starrer Körper wird mit so genannten Bewegungsgleichungen beschrieben. Die jeweiligen Bewegungsfreiheitsgrade der betrachteten Körper werden durch Differentialgleichungen dargestellt. Jeder starre Körper verfügt im dreidimensionalen Raum über jeweils drei Bewegungsfreiheitsgrade für Translation und Rotation. Somit besitzt jeder starre Körper sechs Bewegungsfreiheitsgrade. Als einfaches Beispiel soll hier ein abstraktes Fahrzeug, bestehend aus Fahrzeugkarosserie und den vier Rädern mit den jeweiligen Kopplungen, betrachtet werden. Mit diesem einfachen Modell lassen sich die wesentlichen Bewegungen des Fahrzeugs verdeutlichen, siehe Abb. 2.1. Für das Gesamtsystem erhält man durch die Bewegungsmöglichkeiten der Teilkörper  $(1 + 4) \cdot 6 = 30$  Bewegungsfreiheitsgrade. [Met03]

Wie aus diesem einfachen Beispiel ersichtlich ist, steigt bei genauerer Modellierung, z. B. der Fahrzeugfederung und der Antriebskomponenten, die Anzahl der Systemfreiheitsgrade rapide an. Die Kopplungen zwischen den Differenzialgleichungen sind dabei sehr komplex. Modelle für die Regelungstechnik können daher nur unter großzügigen Vereinfachungen rea-



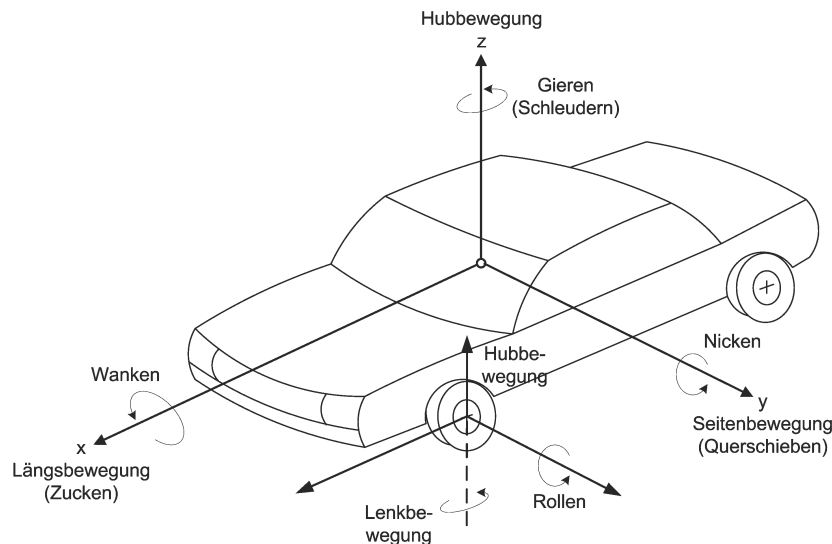


Abbildung 2.1: Fahrzeugbewegungsfreiheitsgrade nach [Mit95]

lisiert werden. Zur Entwicklung einer komplexen Regelung und einer Grundparametrisierung sind aber hinreichend genaue und komplexe Modelle unabdingbar. Hieraus ergibt sich ein Zielkonflikt bei der Entwicklung, der entsprechend der Aufgabenstellung zu lösen ist.

## 2.2 Antriebsstrangdynamik

Bei der Antriebsstrangsimulation steht die Verbesserung der Fahrleistung und des Verbrauchs im Vordergrund. Des Weiteren werden mechanische und akustische Schwingungen des Antriebsstrangs hinsichtlich Fahrkomfort untersucht. Die rasante Entwicklung der Simulationstechnik erlaubt, dynamische Drehmomenteffekte des Antriebsstrangs, die auf die Fahrzeugbewegung wirken, zu untersuchen.

In der Abb. 2.2 ist das komplette Antriebsstrangmodell eines Fahrzeugs dargestellt. Bei der Entwicklung der einzelnen Komponenten werden die entsprechenden Teilgebiete Mechanik, Elektrik/Elektronik, Thermodynamik sowie Hydraulik berücksichtigt. Für Detailbetrachtungen stehen in den Arbeiten von [Wil98, Haj02, Koc01, Tsc02] sehr ausführliche Modelle zur Verfügung. Die detaillierten dynamischen Effekte lassen sich jedoch oftmals nur durch *Finite-Elemente-Modelle (FEM)* realisieren, die nur mit einem großen Rechenaufwand zu berechnen sind. Somit idealisiert man die Modelle für die Echtzeitsimulation und beschreibt das dynamische Verhalten komplexer Bauteile oftmals mittels Kennlinien oder Kennfeldern. Ein Problem bei der Simulation stellt dabei die Beschaffung der Daten dar.

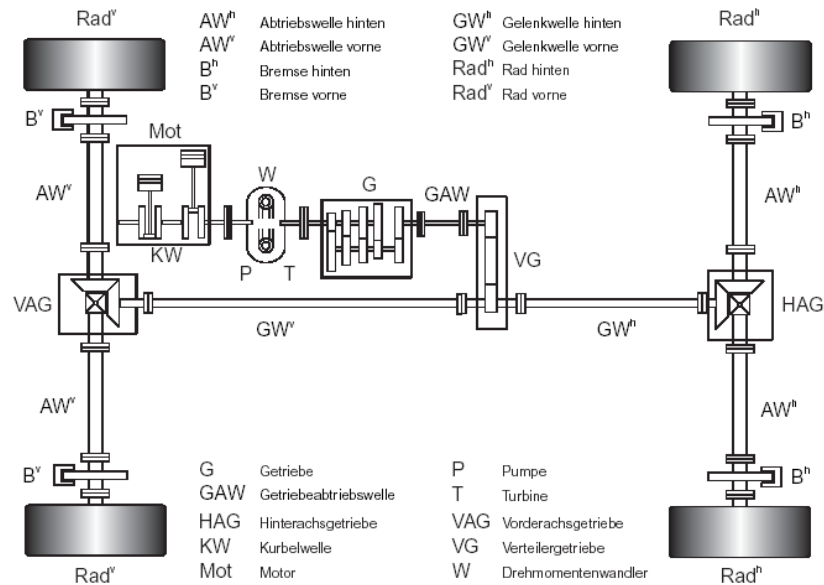


Abbildung 2.2: Aufbau eines allradgetriebenen Kraftfahrzeugs nach [Tsc02]

## 2.3 Fahrzeug – Fahrer – Umwelt

Für die rechnergestützte Entwicklung im Automobilbereich ist die Modellierung des Systems „Fahrzeug – Fahrer - Umwelt“ von zentraler Bedeutung. Denn nach [Mit90] wird das Fahrverhalten eines Kraftfahrzeugs definiert als die Reaktion auf das Lenken des Fahrers, auf das Beschleunigen und Verzögern über Fahr- und Bremspedal während einer Kurvenfahrt sowie die Reaktion auf äußere Störungen. Der Fahrer ist in diesem System für die Fahrzeugführung verantwortlich. Bei der Entwicklung von Regelungen muss der Fahrer mitbetrachtet werden, es entsteht somit ein geschlossener Regelkreis, wie in der Abb. 2.3 dargestellt. [Met03]

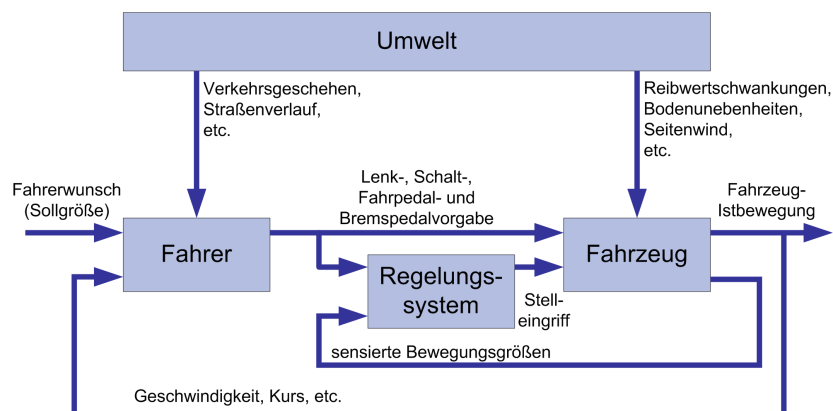


Abbildung 2.3: Regelkreis Fahrzeug – Fahrer - Umwelt nach [SZ03]

In den letzten Jahren hat sich gezeigt, dass die Fahrsicherheit und -stabilität eines Auto-

mobils nicht nur von den fahrdynamischen Eigenschaften abhängt. Vielmehr muss der Fahrer in die Betrachtung mit einbezogen werden. Die menschliche Zuverlässigkeit und Reaktionsfähigkeit spielen für diese Betrachtungen eine große Rolle. Für das Auslegen und Testen von Regelungssystemen in *Model-in-the-Loop-Simulationen* (*MiL-Simulationen*), *Software-in-the-Loop-Simulationen* (*SiL-Simulationen*) und *Hardware-in-the-Loop-Simulationen* (*HiL-Simulationen*) sind daher Fahrermodelle zur Beurteilung und Automatisierung notwendig. Die Modellierung des Fahrers birgt große Unsicherheiten, da die Eigenschaften jedes Fahrers stark variieren. Aus diesem Grund wird oftmals bei der Untersuchung der Fahrer nicht berücksichtigt, man spricht dann von einer *Open-Loop-Simulation*. Wird der Fahrer im Modell mit berücksichtigt, spricht man von einer *Closed-Loop-Simulation*.

## 2.4 Regelungsentwicklung

Die Steuerungs- und Regelungstechnik bildet das verbindende Element verschiedener Fachrichtungen. Durch Abstraktion des Systems von physikalischen und gerätetechnischen Gegebenheiten wird es zum eigenständigen Fachgebiet. Sie beschäftigt sich insbesondere mit Entwurfsmethoden für allgemeine Steuerungs- und Regelungssysteme.

Das Problem der Reglerentwicklung für den Automobilbereich soll hier anhand der Automatikgetriebesteuerung dargelegt werden. Heutige Automatikgetriebe werden elektronisch-hydraulisch geregelt, womit viele Freiheitsgrade bei der Reglerauslegung bestehen. Des Weiteren wird aufgrund von Komfort-, Sicherheits- und Umweltaanforderungen die Getriebesteuerung über ein Bussystem mit anderen Steuergeräten des Antriebsstrangs und der Fahrdynamik, wie z. B. das ESP-Steuergerät, verbunden. Mit diesen Erweiterungen geht jedoch eine Erhöhung der Komplexität der Regelung und damit des Entwicklungsaufwands einher. Für die Reglerauslegung der Automatikgetriebe muss die Dynamik des Antriebsstrangs modelliert werden, wobei je nach Modellierungstiefe Bereiche wie beispielsweise Mehrkörperdynamik, Kontaktmechanik und Strömungsmechanik beachtet werden müssen. Bei Betrachtung der transienten Schaltvorgänge treten zudem Unstetigkeiten sowohl in den mechanischen wie auch in den hydraulischen Modellen auf. Aufgrund dieser Charakteristik können für die Reglerauslegung keine linearen Ansätze gewählt werden. Vielmehr müssen Ansätze der Methoden der nichtlinearen Optimierung und Reglerauslegung herangezogen werden [Haj02].

Für weitere komplexe Steuerungen und Regelungen im Automobilbereich gelten ähnliche Voraussetzungen. Des Weiteren werden entwicklungsbedingt die Steuerungen und Regelungen separat entwickelt. Gegenseitige Beeinflussungen bei der Kopplung über Bussysteme können erst sehr spät betrachtet werden. Daher bestehen Bemühungen, schon frühzeitig mit *Rapid Prototyping*, siehe Abschnitt 3.5.4, den Reglerverbund zu testen. Hierfür müssen seitens der

Simulationstechnik entsprechende Lösungen entwickelt werden, siehe Abschnitt 2.5. Ein weiteres Problem der Reglerentwicklung ist die Parametrisierung des Modells, die heute größtenteils noch immer auf experimenteller Basis erfolgt [Haj02].

## 2.5 Simulationstechnik

Die Simulationstechnik hat sich in den letzten Jahren zu einer eigenständigen Fachdisziplin entwickelt. Dabei werden für die entsprechenden Fachrichtungen spezielle Simulationsmethoden bereitgestellt. Simulation ist im engeren Sinne die Nachahmung realer Prozesse mittels Computern auf der Grundlage mathematischer Modelle [KN98]. Es ist sinnvoll, die Simulation in mehreren Phasen zu betrachten:

- Modellbildung,
- Parameterbestimmung,
- Untersuchung und
- Auswertung.

Bei der rechnergestützten Simulation eines technischen Systems stellt die Modellbildung den ersten und zugleich wichtigsten Schritt dar. Dabei handelt es sich um eine Abstraktion der physikalischen Wirklichkeit, die meistens nur mit Erfahrung und Intuition des Ingenieurs erfolgreich gelöst werden kann [Haj02]. Dies erfolgt mit Hilfe der Systemtheorie, die für die Modellierung spezielle Methoden und Werkzeuge zur Verfügung stellt.

Für die Simulation dynamischer Systeme sind mathematische Modelle notwendig. Sie lassen sich stets durch algebraische Gleichungen, Differentialgleichungen oder eine Kombination aus beiden, den so genannten *DAE-Systemen* (differential-algebraic equation), bilden. Dies gilt auch dann, wenn zunächst grafische Beschreibungssprachen gewählt wurden. Bei der Modellierung ist eine Balance zwischen hoher Genauigkeit des Modells einerseits und der Forderung nach geringem Rechenaufwand und einer überschaubaren Modellstruktur andererseits zu achten. Hieraus wird ersichtlich, dass für die Modellierung eine genaue Spezifikation der Anforderungen an das Modell notwendig ist. Dabei ist zu beachten, dass die Simulationstechnik immer einen iterativen Prozess darstellt.

Die Modellbildung liefert im wesentlichen nur die Struktur des Modells. Um eine hinreichend gute Übereinstimmung des mathematischen Modells mit dem Original zu gewährleisten, ist deshalb das Modell noch zu parametrisieren. Bei einfachen Modellen ist die Wahl geeigneter Parameter leicht zu realisieren. Jedoch steigt die Komplexität bei der Parameterbestimmung (Parameteridentifikation) bei komplexen Systemen und bei Trenduntersuchungen

(das System existiert in der Wirklichkeit noch nicht) sehr schnell an. In solchen Fällen sollte mit innovativen Ansätzen und hochstrukturierten Daten gearbeitet werden. Erst auf der Basis eines in Struktur und Parametern bekannten mathematischen Modells ist es sinnvoll, eine rechnergestützte Simulation des interessierenden Prozesses anzustreben.

Die rechnergestützte Untersuchung des Modells stellt die „eigentliche“ Simulation dar. Sie beinhaltet die Auswahl von Testfunktionen, mit dem das zu simulierende System angeregt wird, sowie die entsprechende Darstellung der Ergebnisse in aussagekräftiger Form. Für die rechnergestützte Simulation sind dazu numerische Lösungsverfahren anzuwenden. Selbstverständlich kann bei der Untersuchung auch die Echtzeitfähigkeit im Vordergrund stehen, was spezielle numerische Verfahren und Simulationssysteme erfordert. Nach [KN98] ist die Untersuchungsphase ein „selbststeuernder Lernprozess“. Bei der Verifizierung des Modells ergeben sich häufig Ungereimtheiten, die in Folge von Modellierungs-, Parameterfehlern oder numerischen Fehlern auftreten. Eine nochmalige Bearbeitung der entsprechenden Phasen wird dadurch notwendig. Für die Aussagekraft eines Modells kommt demnach der Verifizierung eine bedeutende Rolle zu.

Wie aus Abschnitt 2.4 ersichtlich ist, wird eine Simulation bis hin zur Echtzeitsimulation eines Gesamtfahrzeugs für Entwicklung, Test und Optimierung notwendig. Dazu stehen MiL-, SiL- und HiL-Simulation zur Verfügung, siehe Abschnitt 3.5. Bei der Erstellung von Modellen mit hoher Güte sind die in den Abschnitten 2.1, 2.2 und 2.3 aufgezeigten Probleme zu berücksichtigen. Unter diesen Voraussetzungen entsteht ein nichtlineares Modell. Wegen schnellen und langsamen Systemteilen des Modells ist das zu erwartende Systemverhalten „steif“. Außerdem treten im Modell Unstetigkeiten auf, die bei der Modellierung und Simulation beachtet werden müssen. Aus diesen Gründen wurden bisher für die Echtzeitsimulation nur Gesamtfahrzeugmodelle mit einem hohen Idealisierungsgrad verwirklicht. Diese analytischen Modelle erlauben es jedoch nicht, bestimmte Parameter für den Steuergerätetest zur Verfügung zu stellen. Es ist daher sinnvoll, auf einen objektorientierten physikalischen Ansatz überzugehen. Dazu wird in dieser Arbeit mit der Modellierungssprache *Modelica* und dem Simulationssystem *Dymola* ein komplexes Gesamtfahrzeugmodell erstellt. *Dymola* verfügt außerdem über leistungsfähige Algorithmen zur Aufstellung und Lösung von DAE-Systemen. Zur Echtzeitsimulation wird das am DLR zur Verfügung stehende Echtzeitsystem *xPC Target* von *The Mathworks* eingesetzt.

## 3 Stand der Technik mit Ausblick

In diesem Kapitel wird basieren auf der bestehenden Literatur der Stand der Technik von elektronischen Steuerungen, Regelungen und der Simulationstechnik im Automobilbereich dargestellt. Im Fokus der Betrachtung stehen die Antriebs- und Fahrdynamikregelsysteme. Der weite Bereich der elektronischen Systeme im Fahrzeuginnenraum und der Komfortfunktionen wird hier nicht betrachtet. Außerdem wird ein Ausblick auf die zur Zeit in der Forschung und Entwicklung befindlichen elektronischen Systeme gegeben. Anschließend wird kurz die Bedeutung der Elektronik in der Automobilindustrie beschrieben. Zum Schluss des Kapitels wird die Wichtigkeit der Simulationstechnik in der Fahrzeugentwicklung aufgezeigt. Dabei wird der Softwareentwicklungsprozess für Steuergeräte beschrieben und die Wichtigkeit des Testens dargestellt. Die für weitere Innovationen notwendigen Entwicklungen im Automobilbereich und die sich daraus ergebende ganzheitliche, multidisziplinäre und genaue Modellierung des Systems „Kraftfahrzeug“ werden verdeutlicht.

Es ist jedoch schwierig, den genauen Stand der Technik abzubilden. Denn viele Konzerne halten aus marktpolitischen Gründen die wichtigsten und interessantesten Ergebnisse der Forschung geheim. Aus diesen Gründen wird oftmals die realisierte Funktion beschrieben, aber genaue Modelle oder Regelungsverfahren werden nicht veröffentlicht. Deshalb wurde in dieser Arbeit auf verschiedene Publikationen wie Diplomarbeiten, Dissertationen und Artikel aus der Fachpresse zurückgegriffen.

### 3.1 Fahrdynamikregelungen

In heutigen Kraftfahrzeugen kommen verschiedene Fahrdynamikregelungssysteme zum Einsatz. Sie lassen sich grundsätzlich in vier Kategorien unterteilen:

1. Bremshilfen zum effizienten Verzögern des Fahrzeugs bei gleichzeitiger Erhaltung der Lenkbarkeit,
2. Antriebsschlupf-Regelungssysteme zur Verringerung von Stabilitätsverlusten infolge durchdrehender Antriebsräder,

3. Fahrdynamik-Regelungssysteme zur zusätzlichen Regelung der Fahrzeugquerdynamik und
4. Aktive Aufbau-Regelungssysteme zur Horizontierung des Fahrzeugaufbaus.

Bis auf das aktive Aufbau-Regelungssystem sind alle Regelungssysteme dem Fahrer unterlagert und stellen eine Assistenzfunktion bei kritischen Fahrsituationen dar. Mit Hilfe von Sensoren werden die zum Regeln benötigten Größen erfasst und bei fahrkritischen Situationen wird über Aktuatoren ein Stelleingriff in das System „Fahrzeug“ vorgenommen, siehe Abb. 2.3. Selbstverständlich bleiben alle Funktionen des Fahrers dominierend erhalten. Im Folgenden werden einige fahrdynamische Regelungssysteme, die heute in der Fahrzeugtechnik etabliert sind, beschrieben. Eine Einführung in fahrdynamische Regelungssysteme und das Einspurmodell bietet die Arbeit von [Met03].

### 3.1.1 Antiblockiersysteme

*Antiblockiersysteme (ABS)* sind Regeleinrichtungen im Bremssystem, die das Blockieren der Räder beim Bremsen verhindern und damit die Lenkbarkeit und die Fahrstabilität erhalten [BOS02]. Für Gummireifen gilt nicht die klassische Reibungstheorie, sondern der Reibwert ist eine Funktion der Gleitgeschwindigkeit (Radschlupfkurve). Die Radumfangskraft hängt wiederum von diesem Reibwert ab. Wird beim Abbremsen der Bremsdruck erhöht, nimmt der Bremsschlupf (die Differenz zwischen Radumfangs- und Fahrzeuggeschwindigkeit) zu. Bis zum Erreichen der maximalen Umfangskraft bei ca. 0,1–0,2 des Bremsschlupfes nimmt die Radumfangskraft entlang der so genannten Radbremsschlupfkurve zu, es liegt ein stabiler Zustand vor. Bei weiterer Erhöhung der Bremskraft vergrößert sich der Bremsschlupf und die Radumfangskraft nimmt bis zum Stillstand des Rades ab. In diesem Bereich befindet sich die Bremsung im instabilen Bereich, die Radumfangkräfte lassen sich am Fahrzeug nicht mehr stabil einstellen. Das negative Bremsmoment am Rad ist betragsmäßig größer als das Produkt aus Radumfangskraft und Radradius und bewirkt damit eine negative Winkelbeschleunigung des Rades. Dies führt ohne ABS zu einem Abbremsen des Rades bis zum Stillstand („Blockieren“) in wenigen Zehntelsekunden. Bei blockierten Rädern bedingt die geringere Radumfangskraft einen längeren Bremsweg, und außerdem ist das Fahrzeug nicht lenkbar.

Um das Rad im stabilen Bereich und mit maximaler Verzögerung abzubremsen, wird daher das ABS eingesetzt. Der ABS-Regelkreis ist in der Abb. 3.1 dargestellt. Der Raddrehzahlensensor überwacht den Bewegungszustand des Rades. Tritt an einem Rad eine „Blockierneigung“ auf, gibt die ABS-Regelung Befehle an die Magnetventileinheit, den Aufbau des Radbremssdruckes zu stoppen bzw. diesen Druck abzubauen, bis die „Blockiergefahr“ beseitigt ist. Damit

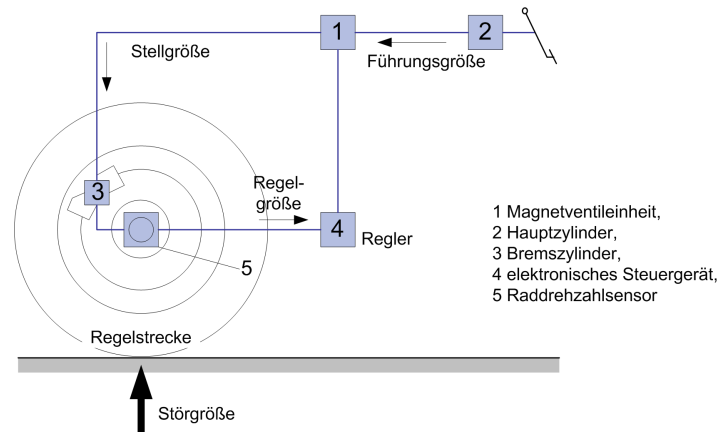


Abbildung 3.1: ABS-Regelkreis nach [BOS02]

das Rad nicht ungebremst bleibt, muss der Bremsdruck dann erneut wieder aufgebaut werden. Diese Regelung findet heute für jedes Rad einzeln und im Millisekundenbereich statt. Ziel einer ABS-Regelung ist es, Spurhaltung und Lenkfähigkeit bei maximaler Verzögerung zu gewährleisten.

Als ABS-Funktionserweiterungen mit Vernetzungen zu anderen Systemen sind unter anderem hinzugekommen [Gau00]:

- Antriebsschlupfregelung, elektronische Differentialsperre, etc.
- Motorschleppmomentregelung als Subfunktion der Antriebsschlupfregelung
- elektronische Bremskraftverteilung
- erweiterte Signalanalyse mit selektiven Bremseingriffen
- Kurvenbremsregelung, etc.

Mit der Einführung des Antiblockiersystems läutete *Bosch* das Zeitalter der Radschlupfregelungen ein. Heutzutage kommen viele ABS-Systeme von unterschiedlichen Herstellern in Automobilen zum Einsatz [BOS02, TEV]. Dabei wurden in den letzten Jahren Sensoren, Aktuatoren und Regelungssysteme weiter entwickelt. In der Serienproduktion ist das ABS Stand der Technik. Es hat einen erheblichen Sicherheitsgewinn für kritische Fahrsituationen gebracht.

### 3.1.2 Antriebsschlupfregelung

Besonders bei glatter Fahrbahn ist der Fahrer oft überfordert, das Gaspedal so dosiert zu betätigen, dass ein Durchdrehen der Räder vermieden wird. Bei den Problemen des Anfahrens und Beschleunigens unterstützt die *Antriebsschlupfregelung (ASR)* den Fahrer durch



Regelung des Antriebsmoments an den Rädern. Entscheidend hierbei ist wiederum der Kraftschluss zwischen Fahrbahn und Reifen, dessen Charakteristik ähnlich der Bremsschlupfkurve beschreibbar ist. Wird das Antriebsmoment beim Anfahren oder Beschleunigen vom Fahrer zu groß gewählt, so erhöht sich die Drehzahl der Antriebsräder durch das wachsende überschüssige Drehmoment sehr schnell, so dass sich der Kraftschluss infolge durchdrehender Räder vermindert [BOS02, Met03]. Die ASR ist eine Erweiterung des ABS und nutzt dessen Komponenten. Übersteigt das vom Fahrer geforderte Antriebsmoment das physikalisch maximal übertragbare Moment, so dreht mindestens ein Antriebsrad durch. Dies wird durch einen Geschwindigkeitsvergleich mit einem nicht angetriebenen Rad erkannt. Der im ABS-Steuergerät integrierte ASR-Bremsregler baut dann an diesem Rad ein Bremsmoment auf. Drehen beide Antriebsräder durch, greift der ASR-Motorregler ein und reduziert das Motormoment (Beeinflussung von Drosselklappe, Einspritzmenge und Zündzeitpunkt über den Datenbus zum digitalen Motormanagement), bis nur noch ein Rad abgebremst ist [Gau00]. Durch die damit verbundene Erhöhung der Fahrstabilität bei beschleunigter Kurvenfahrt und/oder bei wechselnden Straßenreibungswerten wird eine Steigerung der aktiven Fahrsicherheit erreicht. Die ASR greift im Gegensatz zum ABS selbstständig in das Bremssystem ein.

Die Antriebsschlupfregelung ist heute in der gehobenen Mittelklasse und bei stark motorisierten Fahrzeugen Serienausstattung. Der Fahrer hat aber die Möglichkeit, das ASR jederzeit zu deaktivieren. Durch den Einzug von ESP, siehe Abschnitt 3.1.4, in allen Fahrzeugklassen setzt sich auch die Antriebsschlupfregelung immer mehr durch. Die Verbesserungen der Regelungsstrategien und der Technologien werden von unterschiedlichen Firmen weiter vorangetrieben.

### 3.1.3 Elektronische Bremskraftverteilung und Bremsassistent

Die *elektronische Bremskraftverteilung (EBV)* verhindert durch einen angepassten Bremsdruckaufbau ein Überbremsen der Hinterräder. Bisher war es üblich, an der Hinterachse einen Bremskraftbegrenzer zu verwenden, der ein Überbremsen der Hinterräder verhinderte. In der Praxis führte dies dazu, dass aus Sicherheitsgründen der hydraulische Bremsdruck für die Hinterräder viel zu niedrig eingestellt werden musste und somit Bremsweg „verschenkt“ wurde [Gau00].

Bei der EBV handelt es sich um eine *Differenzschlupfregelung*, die in das ABS-Steuergerät integriert wird und die herkömmliche Bremskraftbegrenzung ersetzt. Durch die EBV können nun die Hinterräder stärker zur Abbremsung des Kraftfahrzeugs herangezogen werden und somit ohne Verlust der Fahrzeugstabilität den Bremsweg bei einer Vollbremsung erheblich verkürzen. Der Sicherheitsgewinn der EBV zeigt sich besonders bei der Abbremsung in der Kurve.

Im Zusammenhang mit dem ABS hat sich herausgestellt, dass der Regler „Fahrer“ oft bei Notbremsung das Bremspedal mit zu wenig Kraft beaufschlagt oder sogar bei einsetzendem ABS die Kraft verringert. Damit wird die dem Fahrzeug zur Verfügung stehende Bremskraft nicht optimal ausgenutzt. Darum wurde das *Bremsassistent-System (BAS)* 1996 von Mercedes-Benz entwickelt [Gau00]. Das elektronische BAS verkürzt in Notsituationen durch Bereitstellung der vollen Bremskraft für das ABS den Bremsweg. Ein zentrales Merkmal des BAS ist die Sensierung der Pedalgeschwindigkeit im Bremskraftverstärker. Bei Notbremsungen mit schneller Pedalbetätigung wird die Funktion des BAS ausgelöst. Über eine elektronische Regelung des Unterdruckbremskraftverstärkers wird der Bremsdruck auf sein Maximum erhöht. Wenn das Bremspedal nachdrücklich mit weniger Kraft beaufschlagt wird, d. h. keine Notbremsung mehr notwendig ist, schaltet sich über einen speziellen Löschschalter die Bremskraftverstärkung ab.

### 3.1.4 Fahrdynamikregelung

Die *Fahrdynamikregelung (FDR)*, bekannter unter dem Begriff *elektronisches Stabilitätsprogramm (ESP)*, verbessert und erweitert die Vorteile der Antiblockier- und Antriebsschlupfsysteme in der aktiven Fahrsicherheit in folgenden Punkten [BOS02]:

- Aktive Unterstützung des Fahrers auch in kritischen querdynamischen Situationen
- Erweiterte Fahrstabilität und verbessertes Fahrzeugverhalten, Spur- und Richtungstreue
- Elektronische Bremskraftverteilung
- Erweiterte Signalanalyse (ABS plus) mit selektiven Bremseingriffen im Grenzbereich in allen Betriebszuständen wie Vollbremsung, Teilbremsung, Freieren, Antrieb, Schub und Lastwechsel, auch bei extremen Lenkmanövern (Angst- und Panikreaktionen) und damit drastische Reduzierung der Schleudergefahr
- Verbesserte, situationsabhängige Nutzung des Kraftschlusspotentials bei ABS/ASR und dadurch Bremsweg-/Traktionsgewinne sowie verbesserte Lenkbarkeit.

Neben den Reifenlängskräften, die im Wesentlichen für Bremsung und Beschleunigung ausschlaggebend sind, hängen auch die Reifenquerkräfte (Seitenführungskräfte), die das Fahrzeug während der Kurvenfahrt abstützen, vom Radschlupf ab [BOS02, Met03]. Das Eigenlenkverhalten des Fahrzeugs charakterisiert das Fahrverhalten in der Kurve und hängt ebenfalls vom Radschlupf ab. In diesem Zusammenhang wird als gute Spurtreue bezeichnet, wenn das Fahrzeug einer Fahrspur folgt, die präzise mit dem Lenkwinkelverlauf übereinstimmt. Gewährleis-

tet wird die Spurtreue für Reifenquerkräfte, die deutlich unterhalb des Kraftschlusspotentials zwischen Reifen und Fahrbahn liegen. Der Giergeschwindigkeitsverlauf (Drehung um die Fahrzeughochachse) entspricht dann dem des Lenkwinkels. [BOS02]

Die im ESP enthaltene Giermomentregelung erkennt, ob sich das Fahrzeug noch stabil auf dem vom Fahrer vorgegebenen Kurs bewegt. Dazu kommen beim ESP zusätzlich zu den ABS- und ASR-Sensoren (siehe Abschnitt 3.1.2) Lenkwinkel- und Bremsdrucksensoren sowie ein Drehraten- und Querbeschleunigungssensor hinzu, die die Fahrvorgaben und die tatsächliche Fahrzeugbewegung erfassen. Das ESP vergleicht permanent den Soll- und Istkurs des Fahrzeugs. Registriert das ESP eine Abweichung, erzeugt es durch aktives Bremsen an einzelnen Rädern ein Giermoment, das einem Unter- oder Übersteuern entgegen wirkt [Gau00]. So wird schon im Ansatz verhindert, dass das Fahrzeug ins Schleudern gerät. In der Abb. 3.2 werden die fahrstabilisierenden Maßnahmen bei einem unter- und übersteuernden Fahrzeug dargestellt. Beim Untersteuern wird das kurveninnere Hinterrad, beim Übersteuern das kurvenäußere Vorderrad abgebremst.

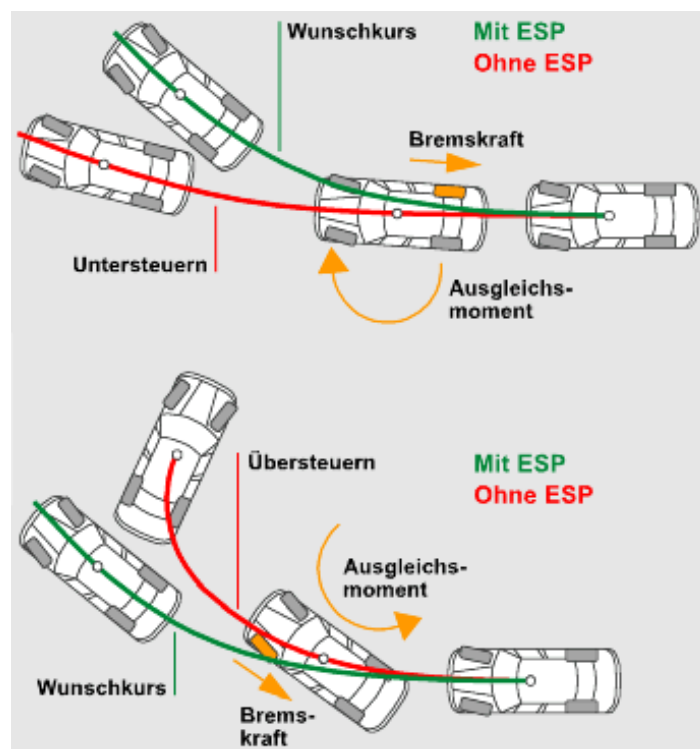


Abbildung 3.2: Fahrstabilisierende Maßnahmen mit ESP [TEV]

Der breiten Öffentlichkeit bekannt ist das ESP durch die Fahrversuche mit der *A-Klasse* von *Mercedes-Benz* geworden, wodurch es Einzug in die heutige Serienproduktion hielt. Die allgemeine Expertenmeinung (so [Gau00, SBT<sup>+</sup>03]) geht davon aus, dass die Unfallzahlen mit ESP deutlich zurückgehen werden. Ein weiteres Potential bietet unter Verwendung von

besseren Sensoren und Aktuatoren die Beeinflussung der Feder- und Dämpfereigenschaften, siehe Kapitel 3.1.7.

### 3.1.5 Aktive Fahrwerke

Die Auslegung des konventionellen Fahrwerks eines Fahrzeugs unterliegt einem klassischen Zielkonflikt. Bei der Abstimmung muss festgelegt werden, ob das Fahrwerk eher in Richtung Fahrkomfort oder in Richtung Fahrdynamik (Handling) optimiert werden soll. Betrachtet man dabei die Fahrwerksparameter der Federung und Dämpfung, sind die Parameter des Systems hinsichtlich eines harmonischen Fahrverhaltens abzustimmen. Erhöht man die Komplexität des Fahrwerks, führt dies zu höheren Kosten. Dennoch kann so bei gestiegenem Komfort ein besseres Handling erreicht werden. Der Zielkonflikt bleibt allerdings bestehen. [Som00]

Ein entscheidender Schritt in Richtung Optimum kann somit nur mit einem Fahrwerk erreicht werden, das bedarfsabhängig und adaptiv die richtigen Parameter zur Verfügung stellt. Mercedes Benz entwickelte 1998 das erste aktive Fahrwerk *Active Body Control (ABC)*. Inzwischen sind weitere aktive Fahrwerke von unterschiedlichen Herstellern, wie z. B. das *Electronic Suspension System (ESS)* von Continental Tevis und das *Dynamic Drive* von BMW, auf dem Fahrzeugmarkt. Mit diesen Systemen werden maßgebliche Parameter fahrsituationsabhängig eingestellt und unterschiedliche Beladungszustände berücksichtigt.

Die elektronischen Dämpfersysteme erkennen über Vertikalbeschleunigungssensoren den Fahrzeug- und Fahrbahnzustand und adaptieren darauf entsprechend ihre Dämpferraten (Schwellwertregler). Durch eine querbeschleunigungs- oder lenkwinkelgeschwindigkeitsabhängige Dämpferverhärtung kann das dynamische Wanken verringert und die Agilität erhöht werden. Die Nickbewegung lässt sich ebenfalls durch Auswertung des Bremsdrucks dämpfen. Eine Weiterentwicklung stellt das semiaktive Skyhook-System dar. Bei diesem Ansatz werden die Dämpferkräfte bei jedem Rad individuell so eingeregelt, dass sich der Aufbau verhält, als sei er mit einem Haken am Himmel (Skyhook) befestigt. Für das Skyhook-System werden zusätzliche Vertikalbeschleunigungssensoren an den Rädern und am Aufbau notwendig. Die Vorteile dieses Systems liegen in einer besseren Aufbaustabilität und gleichzeitiger Radbedämpfung. Weiter kann durch eine aktive elektronisch hydraulische Federung eine Niveauregulierung des Fahrzeugs erfolgen. Dabei senkt das Fahrwerk den Aufbau bei höheren Geschwindigkeiten ab, was eine Verbesserung der dynamischen Achslastverteilung nach sich zieht.

### 3.1.6 Adaptive Cruise Control (ACC)

Das europäische Prometheus-Programm wurde 1994 abgeschlossen. Das Forschungsziel bestand darin, den Verkehr bei hoher Wirtschaftlichkeit sicherer, flüssiger und umweltfreund-

licher zu gestalten [Gau00]. Aus diesem Forschungsprojekt sind die adaptiven Geschwindigkeitsregelsysteme entstanden. Im allgemeinen werden diese Systeme *Adaptive Cruise Control* (ACC) genannt, wobei produktspezifische Bezeichnungen variieren können, wie z. B. ADR, AICC oder ICC.

Bei dem ACC handelt es sich um ein Assistenz-System zur Regelung des Abstandes und der Relativgeschwindigkeit zwischen zwei hintereinander fahrenden Fahrzeugen. In der Grundfunktion basiert das ACC auf dem „konventionellen“ Fahrgeschwindigkeitsregler (Tempomat). Das ACC kann darüber hinaus die Geschwindigkeit selbständig durch Brems- und Motoreingriffe automatisch an die Verkehrsbedingungen anpassen. Es ist in der Lage ein vorausfahrendes langsames Fahrzeug zu erkennen und dementsprechend die Fahrzeuggeschwindigkeit selbsttätig durch eine automatische Bremsung zu verringern, so dass ein sicherer Abstand zu dem vorausfahrenden Fahrzeug eingehalten wird. Bei Geschwindigkeitsänderungen des vorausfahrenden Fahrzeugs werden die eigene Geschwindigkeit und der Abstand entsprechend angepasst. Sobald das vorausfahrende Fahrzeug den „Regelbereich“ vor dem ACC-Fahrzeug verlässt, beschleunigt das ACC auf die vom Fahrer eingestellte Wunschgeschwindigkeit bei freier Fahrt. [Gau00]

Das ACC verwendet die im elektronischen System ESP enthaltenen Funktionen (z. B. ABS und ASR) mit der Erweiterung um eine kompakte Einheit aus Radarsensor und Steuergerät. Der Radarsensor (LRR – Long Range Radar) erfasst vorausfahrende Fahrzeuge bis zu einem Abstand von 100–120 Metern in einem Winkelbereich von  $\pm 4$  Grad zur Mittelachse.

Das ACC entlastet den Fahrer. Der Fahrer bleibt hierbei jedoch in der Verantwortung für das sichere Führen des Fahrzeuges, d. h. alle Aktionen des Fahrers bleiben vorrangig. Zur Zeit erkennen die ACC-Systeme nicht alle Verkehrssituationen korrekt (z. B. Erkennung von Fußgängern oder Fahrradfahrern), deshalb werden sie überwiegend im oberen Geschwindigkeitsbereich auf Schnellstraßen oder Autobahnen eingesetzt. Die Forschung bei Sensoren und Erkennungsprogrammen schreitet schnell voran, so dass in naher Zukunft die ACC-Systeme auch im Stadtverkehr Anwendung finden werden.

### 3.1.7 ESP II – Fahrdynamik der nächsten Generation

ESP II wird von *Continental Teves* entwickelt und nutzt aktive Lenkeingriffe. Damit bietet es mehr Fahrsicherheit und Fahrkomfort gegenüber herkömmlichen Fahrdynamikregelungen. Die Überlagerungslenkung stellt die bedeutendste Komponente innerhalb des ESP II dar. Sie ermöglicht, zu dem vom Fahrer vorgegebenem Lenkwinkel einen zusätzlichen Lenkwinkel in das System einzubringen. Damit setzt sich der Radwinkel aus dem *Fahrerwinkel* und dem *Überlagerungswinkel* zusammen. Vorteile der Zusatzlenkung sind nach [SBT<sup>+</sup>03]:

- Elektronische Beeinflussung der Lenkübersetzung
- Charakterisierung der Fahrzeugdynamik in unkritischen Fahrsituationen durch individuell abgestimmte harmonische Lenkeingriffe
- Regelung der Fahrzeughorizontaldynamik in kritischen Bereichen in Kombination mit dem ESP-System
- Bremsengriffe werden, wenn überhaupt, erst später notwendig.

Durch Integration der Überlagerungslenkung in die Fahrdynamikregelung ESP II erschließen sich zusätzliche Funktionen. Das System ist in der Abb. 3.3 mit seinen Komponenten dargestellt. Im Folgenden werden die Mehrwertfunktionen des Systems beschrieben.

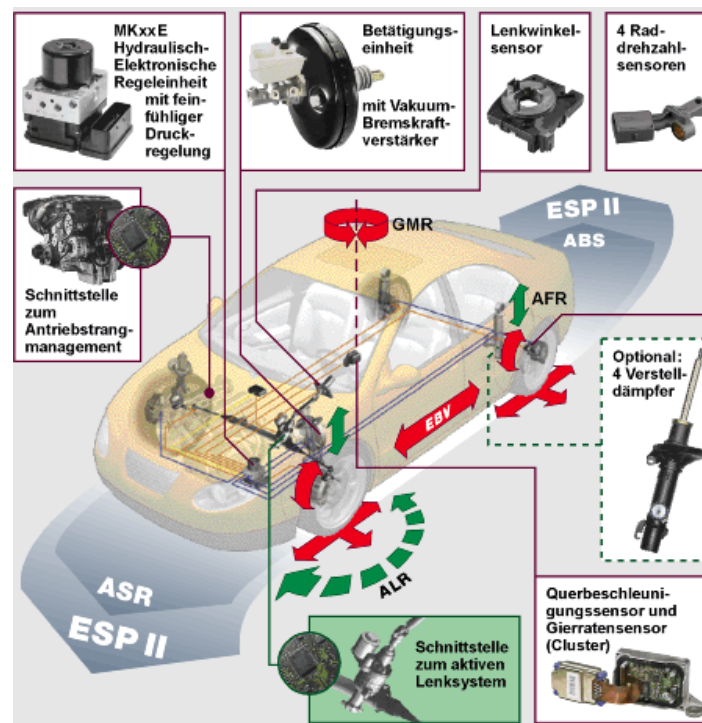


Abbildung 3.3: ESP II – Funktionen und Komponenten [TEV]

### Handling-Support (Vorhaltelenkung)

Aufgrund der elastischen Fahrwerkselemente wie Reifen und Gummilager in Verbindung mit der Fahrzeugträgheit um die Hoch- und Längsachse reagiert das Fahrzeug verzögert auf Lenkimpulse. Zur Verbesserung des Fahrzeugverhaltens berechnet ESP II aus der Fahrzeuggeschwindigkeit, dem Lenkwinkel und der Querbewegung eine Fahrzeug-Sollreaktion. Das Fahrzeug-Sollverhalten wird permanent mit dem Istverhalten verglichen. Tritt zwischen

Soll- und Istverhalten ein Phasenverzug auf, wird vom Steuergerät kurzzeitig ein Vorhalte-  
winkel von der Überlagerungslenkung angefordert. Die Fahrzeugreaktion wird beschleunigt,  
wobei durch den geschlossenen Regelkreis über die Gierbeschleunigung auch veränderte Be-  
ladungszustände, Reifentypen und Verschleißerscheinungen des Fahrwerks kompensiert wer-  
den. [SBT<sup>+</sup>03]

### **Giermomentkompensation**

Bei Bremsvorgängen auf unterschiedlich griffigen Fahrbahnhälften ( $\mu$ -split-Bremsung) wird  
von herkömmlichen Bremssystemen ein Giermoment in Richtung der Seite mit dem höher-  
en Reibwert erzeugt. Bei ESP-Systemen wird durch unterschiedlichen Aufbau des Brems-  
drucks ein entgegengesetztes Giermoment erzeugt, welches zur Fahrstabilität des Fahrzeugs  
beiträgt. Dabei wird das Fahrzeug jedoch nicht mit maximaler Verzögerung abgebremst, wo-  
durch Bremsweg verschenkt wird, der in Notsituationen lebenswichtig ist. Bei ESP II besteht  
zusätzlich die Möglichkeit des elektronischen Lenkeingriffs. Das von unterschiedlichen Brems-  
kräften herrührende Giermoment lässt sich situationsgerecht durch automatisches schnelles  
Gegenlenken kompensieren. Der Fahrer kann dabei weiter in die gewünschte Richtung lenken.  
Durch die schnelle Giermomentkompensation der Zusatzlenkung kann ein größeres Bremsmo-  
ment an den Rädern der Vorder- und Hinterachse aufgebaut werden. Der Bremsweg reduziert  
sich um bis zu 15%. [SBT<sup>+</sup>03]

### **Gierratenregelung**

Mit der Integration der Zusatzlenkung in die Gierratenregelung verfügt ESP II neben dem  
Bremseingriff über eine weitere effektive Möglichkeit, die Fahrzeughorizontaldynamik zu be-  
einflussen. Das Grundprinzip der Gierratenregelung von ESP II baut auf der Standard-ESP-  
Regelung auf. Aus Fahrzeuggeschwindigkeit, Lenkwinkel und Querbewegung wird über  
das sogenannte Referenzmodell eine Soll-Gierrate als Hauptführungsgröße berechnet. Ba-  
sierend auf der Abweichung zwischen gemessener Gierrate und Soll-Gierrate berechnet das  
ESP II ein Korrekturmoment, welches das Fahrzeug auf den Soll-Kurs zwingen soll. Das durch  
die Lenkung erzeugte Korrekturmoment ist dabei größer als das Korrekturmoment der Brem-  
sen. Dadurch wird ein Eingreifen der Bremsen bei unkritischen Manövern nicht notwendig,  
die Fahrdynamik wird erhöht. Bei kritischen Fahrsituationen ist das Eingreifen der Bremsen  
zur Verringerung der Geschwindigkeit jedoch unverzichtbar.

Darüber hinaus können Lastwechselreaktionen erkannt und durch Eingriff der ESP-II-  
Regelung komfortabel kompensiert werden. Ein Überschlagen des Fahrzeugs, z. B. für *Sport*  
*Utility Vehicles (SUV)*, kann durch das Erkennen von schnellen Lenkbewegungen vermieden  
werden.

ESP II bietet außerdem die Möglichkeit, eine Fahrwerksregelungsfunktion mit Verstelldämp-



fern zu integrieren, wodurch der Bremsweg optimiert und das Eigenlenkverhalten des Fahrzeugs beeinflusst wird. Die ESP-II-Regelung stellt einen konsequenten Schritt in Richtung Global Chassis Control dar, siehe Kapitel 3.3. Zurzeit wird die aktive Lenkung im neuen 5er und 6er von BMW eingesetzt. Hierbei werden besonders die Komforteigenschaften der Lenkung bei Geradeausfahrt und Einparken verbessert. Eine gesamtheitliche Entwicklung mit den Fahrdynamikregelungssystemen erfolgt jedoch nur bedingt. Hier bietet der Entwicklungsansatz von ESP II mehr Möglichkeiten für die Erhöhung der Fahrsicherheit. Eine weitere Markteinführung ist in der Fahrzeugoberklasse zu erwarten.

## 3.2 Antriebsstrangdynamik

Beim Antriebsstrang handelt es sich um ein komplexes schwingungsfähiges System. Bei der Entwicklung sind sich zum Teil widersprechende Zielfunktionen wie Agilität und Verbrauch zu berücksichtigen. Der Antriebsstrang besteht aus Verbrennungsmotor, Getriebe und Hilfsaggregaten und ist heute nur mit Hilfe mechatronischer Systeme effizient zu gestalten. Daher hat selbstverständlich die Mechatronik im Antriebsstrang Einzug gehalten. Im Folgenden soll der Entwicklungsstand der Verbrennungsmotoren, Automatikgetriebe und Steuerungs- und Regelungstechnik kurz aufgezeigt werden.

### 3.2.1 Verbrennungsmotoren

Der Entwicklungsstand bei Verbrennungsmotoren ist sehr hoch. In den letzten Jahren standen die hohen Anforderungen an Komfort und Fahrleistung den Forderungen nach Verbrauchsreduzierung und niedrigeren Emissionen gegenüber. Dieser Zielkonflikt ist nur durch konsequente Entwicklung in allen Teilgebieten der Verbrennungsmotoren und mit innovativen Ansätzen zu lösen. Grundsätzlich lassen sich zwei Motorenkonzepte unterscheiden:

- Hubkolbenmotor mit innerer Verbrennung und Fremdzündung (Ottomotor)
- Hubkolbenmotor mit Selbstzündung (Dieselmotor).

Beide Konzepte sollen im Folgenden kurz erläutert werden. Anschließend werden die Potenziale der elektronischen Motorsteuerungen aufgezeigt.

#### Ottomotoren

Der Ottomotor wurde Ende des 19. Jahrhunderts von Nikolaus Otto auf der Basis des erheblich leistungsschwächeren Gasmotors entwickelt und findet heute weite Verbreitung in Kraftfahrzeugen. In heutigen Ottomotoren kommen Leichtbautechnik, elektromechanische Drosselklappe, variable Ventilsteuerung, Aufladung, Mehrventiltechnik und Mehrfachzündung mit



einer gekoppelten Motorsteuerung zum Einsatz. Anhand des *BMW*-Achtzylinder-Ottomotors [BJT<sup>+</sup>01] wird der Entwicklungsstand kurz aufgezeigt. Bei den *BMW-VALVETRONIC*-Motoren kommt keine herkömmliche Drosselklappe mehr zum Einsatz, sondern die Zylinderfüllung wird direkt über eine bedarfsgerechte stufenlose Öffnung der Einlassventile gesteuert. Dadurch reduzieren sich die Verluste beim Ladungswechsel. Durch den Einsatz von Simulation und Leichtbauteilen wurde eine Gewichtsreduzierung des Kolbens und der Pleuelstange erreicht. Dies führt zu geringeren Drehmomentschwingungen und erhöht damit den Komfort. Die Ventile sind derart angeordnet, dass der Ein- und Auslassluftstrom vergrößert wird und sich ein sehr kompakter Brennraum mit optimalem Oberfläche-Volumen-Verhältnis ergibt. Zum Einsatz kommt auch eine stufenlos variable Saugrohranlage, wodurch in jedem Drehzahlbereich die Zylinderfüllung optimiert wird.



Abbildung 3.4: *BMW-VALVETRONIC*-Motor mit Getriebe [BJT<sup>+</sup>01]

Des Weiteren werden Ottomotoren mit Benzindirekteinspritzung (*BDE*) wie beispielsweise von Volkswagen mit den *FSI*-Motoren angeboten. Die verwendeten konstruktiven Mittel zur Optimierung der Verbrennungsmotoren sind nur in enger Zusammenarbeit mit der Softwareentwicklung, siehe Abschnitt 3.4, effektiv einsetzbar. Eine umfangreiche Einführung in die Grundlagen der Verbrennungsmotoren findet man in [Mau03, BOS02].

### Dieselmotoren

Der Dieselmotor wurde 1892 von Rudolf Diesel erfunden. Ursprünglich wurde er mit Pflanzenöl betrieben. Der aus Erdöl hergestellte Dieseldieselkraftstoff wurde erst später massenproduziert und nach ihm benannt. Die Diesel-Technologie hat sich in den letzten Jahren enorm entwickelt. Dabei verbindet der Dieselmotor sehr gute Fahreigenschaften mit geringem Verbrauch. Diese wurden realisiert durch konsequente Umsetzung von:

- Turboaufladung mit variabler Turbinengeometrie und Ladeluftkühlung,
- Direkt- und Mehrfacheinspritzung in den Brennraum,
- Erhöhung des Einspritzdrucks bis auf 2000 bar durch Common-Rail- und Pumpe-Düse-Technik,
- Aluminiumlegierungen in der Zylindergehäusekonstruktion und
- Abgasrückführung.

Selbstverständlich kommen der Motorsteuerung auch beim Dieselmotor entscheidende Aufgaben zur Komfort- und Verbrauchsverbesserung zu. Eine gute Einführung bieten hier [vS02] und [BOS02].

### 3.2.2 Automatikgetriebe

Die Anforderungen an Automatikgetriebe in Bezug auf Komfort und Agilität sowie Umweltverträglichkeit und Fahrspaß steigen. Diese im Widerspruch stehenden Anforderungen verlangen eine an die Fahrsituation angepasste Abstimmung. Deshalb werden moderne Automatikgetriebe mechatronisch gesteuert.

Seit über 30 Jahren werden Automatikgetriebe für Kraftfahrzeuge gebaut und verkauft. Dabei gibt es Automatikgetriebe in unterschiedlichen technischen Ausführungen. Am Weitesten verbreitet sind derzeit die Stufenautomaten auf der Basis von Planetensätzen [Koc01]. In modernen Stufenautomaten werden die verschiedenen Übersetzungsverhältnisse mittels Planetensätzen realisiert, die über kraftschlüssige Kupplungen, Bremsen und Freiläufe verschaltet sind. Die Entwicklung der Automatikgetriebe führte über Dreigang-Automatikgetriebe zu Viergang-Automatikgetrieben, wobei die Schaltstrategien immer komplexer wurden. So ist eine Verdopplung des Softwareumfanges alle zwei bis drei Jahre zu beobachten [Run01]. Heutige Automatikgetriebe werden als Vier-, Fünf-, Sechs- oder sogar als Siebengang-Automatikgetriebe gefertigt. Die Ankopplung an den Verbrennungsmotor übernimmt ein hydrodynamischer Drehmomentwandler mit einer Überbrückungskupplung zur Verbrauchsreduzierung. So wird ein Schaltvorgang ohne Drehmomentunterbrechung möglich. Zunehmende Bedeutung gewinnen *stufenlose Getriebe (CVT)*, bei denen das Übersetzungsverhältnis kontinuierlich in einem Bereich veränderbar ist. Das Übersetzungsverhältnis wird über verstellbare Räder bei An- und Abtrieb, die beispielsweise über ein Band oder eine Kette verbunden sind, realisiert.

Eine weitere Getriebebauart stellt das automatisierte Schaltgetriebe dar. Getriebe dieses Typs entlasten den Fahrer von den Aufgaben Kuppeln und Gangwahl bei gleichzeitiger Beibehaltung des Verbrauchsvorteils von Schaltgetrieben gegenüber Automatikgetrieben. Der

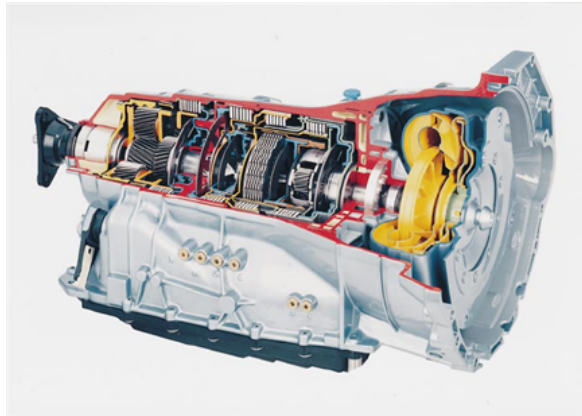


Abbildung 3.5: Das Sechsgang-Automatikgetriebe 6 HP 26 von ZF Friedrichshafen AG [Run01]

Gangwechsel wird mittels Sensorik, Aktuatorik und elektronischer Steuerung automatisiert. Eine gute und kurze Einführung in Automatikgetriebe gibt [DGK01].

### 3.2.3 Antriebsstrangsteuerungen

#### Motorsteuerungen

Die Steuerung und Regelung des Verbrennungsmotors übernimmt das *elektronische Motorsteuergerät (EMS)*. In den letzten Jahren ist der Umfang softwaregesteuerter Funktionen im Steuergerät rapide angestiegen. Des Weiteren wirken andere Steuergeräte auf die Motorsteuerung ein (z. B. ESP-Steuergerät). Die Aufgabe des Motorsteuergerätes besteht darin, auf der Basis von Sensordaten und Algorithmen alle Aktuatoren so zu steuern, dass sich ein bestmöglicher Motorbetrieb bezüglich Verbrauch, Leistung, Abgasemission und Fahrkomfort einstellt. In das Motorsteuergerät werden außerdem Steuerungs- und Regelungsalgorithmen für die Nebenaggregate und Abgasnachbehandlung integriert. Über eine Busschnittstelle kommuniziert das Motorsteuergerät mit anderen Steuergeräten. Man spricht in diesem Zusammenhang auch vom so genannten Motormanagement. In modernen Motormanagementsystemen sind unterschiedlichste Steuerungen und Regelungen enthalten wie z. B.:

- Luft-Kraftstoff-Regelung,
- Last- und Drehzahlregelung,
- Klopfregelung,
- Ladedruckregelung,
- Leerlaufdrehzahlregelung,

- Motorleistungssteuerung sowie
- Zylindergleichstellung.

Bei heutigen elektronischen Motorsteuerungen hat der Fahrer keinen direkten Einfluss auf die eingesetzte Kraftstoffmenge. Vielmehr wird die Kraftstoffmenge durch eine Vielzahl von Einflussgrößen wie beispielsweise Betriebszustand, Fahrerwunsch und Schadstoffemission bestimmt. Selbstverständlich werden beim Motormanagement komplexe Sicherheits- und Diagnosefunktionen integriert. Allgemein lässt sich ein Trend bei der Motoroptimierung hin zu Software-Parametern beobachten, wobei die Eigenschaften des Endprodukts zunehmend die Auslegung des Motormanagement-Systems bestimmt.

### Getriebesteuerung

Der komplette Ersatz von mechanischen Baugruppen wie Freiläufe oder Synchronisierungen durch die Komponente Software zeigt auch auf der Getriebeseite den klaren Trend „Weg von der Mechanik, hin zur Elektronik“ [Run01]. Prinzipiell werden vom *elektronischen Getriebe-steuergerät (EGS)* folgende Funktionalitäten in der Software abgearbeitet:

- Ermittlung des Schaltzeitpunktes,
- Ausführen der Schaltung gemäß den Vorgaben der Schaltpunktermittlung,
- Signalverarbeitung, Überwachung und Diagnose und
- Ansteuerung der Überbrückungskupplung des hydraulischen Wandlers.

In modernen Automatikgetrieben übernimmt die Steuerung des Getriebes eine mechatronische Getriebesteuerung – eine Kombination aus hydraulischer Steuerlogik und integriertem Elektronikbaustein, siehe Abb. 3.6. Elektronik, Sensorik, Aktuatorik sowie Hydraulik sind im Getriebegehäuse untergebracht und speziell aufeinander abgestimmt. Dies ermöglicht eine genauere Steuerung und gleichzeitig eine Gewichtsverringerung sowie eine höhere Zuverlässigkeit. [Run01]

Die Regelung des Schaltungsablaufs wird mit einer Steuerung und überlagerter Regelung realisiert. Prinzipiell ist auch ein geregelter Schaltungsablauf mit einem optimalen Zustandsregler möglich. Die Regelungstheorie kann dabei den Haft-Gleit-Übergang der Schaltelemente berücksichtigen. Selbstverständlich ist das Getriebesteuergerät über ein Bussystem mit dem Motormanagement verbunden und kann so direkt Einfluss auf die Drehmomenterzeugung nehmen. Die Schaltqualität wird zudem durch adaptive Steuerungen zur Kompensation von Fertigungstoleranzen und Laufzeiteffekten erhöht.

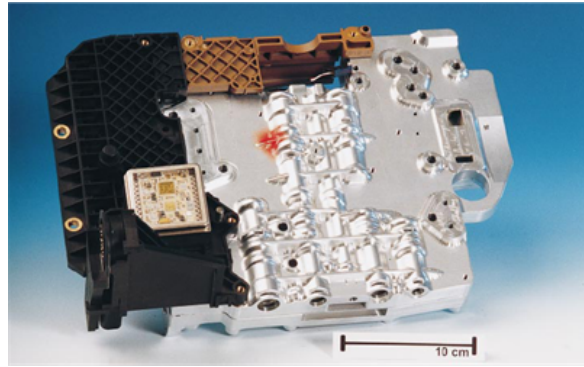


Abbildung 3.6: Das Mechatronik-Modul des 6-HP-26-Getriebes von der *ZF Friedrichshafen AG* [Run01]

### Antriebsstrang-Management

Die Verbesserung des Fahrkomforts und die Reduzierung von Schadstoffemissionen und Verbrauch bedingen eine ganzheitliche Betrachtung des Antriebsstrangs. Bei der Entwicklung von modernen Kraftfahrzeugen werden folglich die Komponenten Motor, Getriebe und Bremse nicht mehr getrennt betrachtet, sondern die Steuerungen und Regelungen werden optimal aufeinander abgestimmt entwickelt. Es entsteht ein so genanntes Antriebsstrang-Management-System, dass die Aufgaben des Antriebstrangs „intelligent“ koordiniert. Dabei ist die Drehmomentstruktur des gesamten Antriebsstrangs bei der ganzheitlichen Reglerauslegung ein entscheidender Faktor. Dieses Antriebsstrang-Management muss auf Basis des für den tatsächlichen Vortrieb bzw. die tatsächliche Verzögerung entscheidenden Summenradmoments aufgebaut werden. Mittels des Antriebsstrang-Managements lassen sich Verbrauch und Agilität verbessern und außerdem durch gezieltes Rückschalten die Bremsanlage entlasten. [JVWW01]

## 3.3 Die Zukunft des Automobils

In diesem Abschnitt werden die zukünftigen Potentiale bei der Automobilentwicklung betrachtet. Im Besonderen werden die fahrdynamischen Regelungssysteme und die Probleme in der Entwicklung der Teilsysteme aufgezeigt.

Die Forderung nach mehr Verkehrs- und Fahrsicherheit sowie Fahrkomfort verlangt nach innovativen Systemlösungen. Hierbei kommt dem Chassis eine zentrale Bedeutung zu, da es wie kein anderer Teil die Fahrsicherheit und den Fahrkomfort des Fahrzeugs bestimmt. Die heutigen Chassis-Subsysteme wie Bremse, Lenkung, Federung/Dämpfung und Reifen sind weitestgehend als eigenständige (Stand-alone-) Produkte konzipiert. Aufgrund zum Teil fehlender standardisierter Schnittstellen besteht bei der Integration einzelner Steuerungs- und Regelungssysteme, wie z. B. die Integration von Klimasteuerung mit der Fenstersteuerung oder

Feder-/Dämpfersteuerung mit dem ESP, zum Gesamtsystem ein enormer Entwicklungsaufwand. Das Gesamtsystem Fahrzeug stößt somit zwangsläufig an seine Kosten-/Nutzengrenze. Diese lässt sich durchbrechen, indem mit der funktionalen Vernetzung der Subsysteme und dem damit einhergehenden Wandel in der Systemarchitektur und Gerätetopologie der nächste große Technologiesprung vollzogen wird. Einen möglichen Ansatz bietet hier das System *Global Chassis Control (GCC)*. Im Folgenden werden die Innovationen für GCC aufgezeigt. Dazu wird die Wichtigkeit der Mechatronik für den Automobilbau, die aktiven Chassis-Systeme und X-By-Wire beschrieben. [SR03]

### 3.3.1 Mechatronik

Die Steuerungs- und Regelungstechnik mit moderner Sensorik, Aktuatorik und digitaler Signalverarbeitung hat einen besonders hohen Anteil an den Innovationen im Kraftfahrzeug. Sie ist integraler Bestandteil von mechatronischen Entwicklungen und erlaubt vielfältige neue Funktionen.

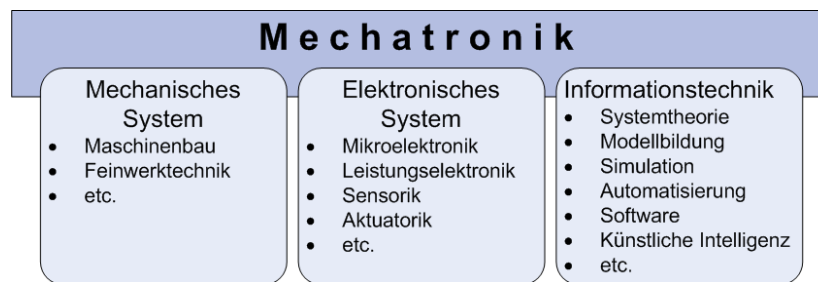


Abbildung 3.7: Mechatronik: Integration verschiedener Disziplinen mit synergetischen Effekten

Die Mechatronik hat in den letzten Jahren mit den ABS- und ESP-Systemen Einzug in den Automobilbau gehalten. Dabei stand die funktionelle Erweiterung des „klassischen Maschinenbaus“, die nur durch die Elektronik erreicht werden kann, bei gleichzeitiger Reduzierung der Kosten im Vordergrund. Mechatronik ist ein interdisziplinäres Gebiet der Ingenieurwissenschaften, das auf den klassischen Disziplinen Maschinenbau, Elektrotechnik und Informatik aufbaut. Zukünftige Innovationen können dabei nur durch einen hohen Integrationsgrad der einzelnen Teilgebiete auf der Systemebene erreicht werden.

Mechatronische Systeme zeichnen sich dadurch aus, dass sich die Integration sowohl auf die Funktion als auch auf die anschließende konstruktive Verschmelzung von Informationsgebern, Stelleinrichtungen und Informationsverarbeitungseinheiten bezieht. Schließlich entstehen Systeme mit adaptiven und autonomen Eigenschaften, die insbesondere bei der Behebung von Fehlern zahlreiche Vorteile bieten.

Bei mechatronischen Systemen im Kraftfahrzeug steht selbstverständlich auch die Über-schaubarkeit und Beherrschbarkeit des gesamten Entwicklungsprozesses im Vordergrund. Bei der ganzheitlichen Betrachtung der Systeme ist eine frühzeitige und über alle Fachdisziplinen gehende Simulation unabdingbar. Hier sind neue Ansätze wie die objektorientierte Modellierung, siehe Kapitel 5, notwendig. Durch die Komplexität und Heterogenität ist eine systematische Vorgehensweise mit rekursiven Phasen nötig. Der Austausch von Informationen der einzelnen Fachdisziplinen im Team muss auf einem hohen Niveau erfolgen.

### 3.3.2 Aktive Chassis-Systeme

#### Bremssysteme

Das Bremssystem muss für GCC-Systeme aktiv ausgelegt werden. Das bedeutet, die Systeme müssen ohne Betätigung des Fahrers Bremskraft aufbauen können. Das Grundprinzip besteht in der hydraulischen Trennung zwischen Bremspedal und Radbremsen. Der Bremswunsch vom Fahrer wird von der Pedaleinheit redundant erfasst. Die Signale werden „by wire“ an das Steuergerät übertragen, siehe Abschnitt 3.3.4, im Steuergerät verarbeitet und entsprechende Bremseingriffe erzeugt, die schließlich mittels Datenübertragung an die einzelnen Radbremsen gelangen. Durch dieses System können Bremsfunktionen allein durch elektronische Regelungen (z. B. ESP) ausgelöst werden.

Bei der *elektrohydraulischen Bremse (EHB)* werden die Bremskräfte noch hydraulisch an die Räder übertragen. Für das System besteht im Notfall (z. B. Ausfall der Elektronik) eine hydraulische Rückfallebene, die den passiven Bremseingriff ermöglicht. Durch diese, im Normalzustand der Bremsanlage, rein elektronische Kopplung von Fahrer und Bremssystem (Brake-By-Wire) ist die Voraussetzung für die funktionale Vernetzung der Anlage über eine geeignete Schnittstelle mit übergeordneten Regelsystemen (z. B. GCC) gegeben.

Das Erweitern bzw. das Hinzufügen von neuen Komponenten führt zu sehr komplexen Bremssystemen, die nur noch schwer in den enger werdenden Einbauräumen zu handhaben sind. Einen weiteren Schritt in die Zukunft verspricht das *elektromechanische Bremssystem (EMB)*. Hier werden die Bremskräfte durch elektromechanische Radbremsaktuatoren direkt an den Rädern erzeugt. Dadurch entfallen die hydraulischen Komponenten. Es entsteht ein so genanntes „trockenes“ Brake-By-Wire-System [SR03]. Aus Sicherheitsgründen muss genügende Redundanz im Bordnetz, im Bussystem und in der Aktuatorik vorhanden sein, da es keine mechanische Rückfallebene gibt. Die EMB benötigt aufgrund ihrer Leistung ein sicheres 42-V-Bordnetz, das zur Zeit noch in der Entwicklung ist.



### Zusatzlenkung

Zur Zeit werden hauptsächlich passive Lenksysteme mit Hilfskraftzuführung eingesetzt. Für GCC-Systeme ist es jedoch notwendig, aktive fremdkrafterregte elektronische Lenksysteme zu entwickeln. Mit diesen Lenksystemen ist eine elektronisch gesteuerte Überlagerung des Lenkwinkels, also eine frei wählbare Lenkübersetzung möglich. Hierbei bietet die Überlagerungslenkung neben der elektrohydraulischen und elektromechanischen Lenkung eine nahezu vollständige GCC-Kompatibilität. Dabei bietet eine Kombination aus Überlagerungslenkung zusammen mit der lenkmomentbeeinflussbaren Servolenkung einen sinnvollen Zwischenschritt in Richtung Steer-by-Wire.

Neben der Grundfunktion der variablen Lenkübersetzung eröffnet sich im Systemverbund von GCC mit der Möglichkeit der aktiven Lenkwinkelüberlagerung ein hohes Potenzial für weitere Funktionen. Gegenüber dem reinen ESP-Bremseingriff lässt sich im Rahmen einer Fahrstabilitätsregelung mit kombiniertem Brems- und Lenkeingriff die Fahrdynamik und Fahrstabilität bei gleichzeitig leichter beherrschbarem Grenzbereich weiter erhöhen [SR03]. Bei der *elektrohydraulischen Lenkung (EHL)* mit manueller Rückfallebene sind im Normalbetrieb die Lenkung und der Lenkaktuator energetisch entkoppelt. Die Übertragung der Lenkvorgabe des Fahrers erfolgt rein elektrisch. Völlig ohne hydraulische Flüssigkeit kommt die *elektromechanische Lenkung (EML)* aus. Auf Basis eines Lenksensors werden die Stelleingriffe rein elektromechanisch erzeugt und ein Lenkgefühl-Simulator generiert dem Fahrer Lenkrückwirkungen. Selbstverständlich müssen die elektronischen Komponenten, wie z. B. Sensoren und Bussysteme, redundant ausgelegt werden, ebenfalls ist ein sicheres 42-V-Bordnetz einzusetzen.

### Feder-/Dämpfersysteme

Bei den Feder-/Dämpfersystemen kann man zwischen passiven, semiaktiven und aktiven Systemen unterscheiden. Diese Systeme unterscheiden sich durch den Kraftstellbereich, den beeinflussbaren Frequenzbereich, ihren Energiebedarf und den Aufwand an Aktuatoren und Sensoren.

Das so genannte „langsam aktive Fahrwerk“ *Active Body Control (ABC)* von Daimler-Chrysler kann alle Bewegungsrichtungen des Fahrzeugaufbaus bis zu einer Frequenz von 5 Hz aktiv beeinflussen. Mittels eines hydraulischen Stellzylinders wird die in Reihe geschaltete (steife) Stahlfeder verstellt. Zu dem in Reihe liegenden Verstellplunger und der Feder ist parallel ein (weicher) passiver Dämpfer angeordnet.

BMW bietet mit dem *Dynamic Drive* ein aktives Feder-/Dämpfersystem zur Beeinflussung der Wankbewegung an. Ein hydraulischer Schwenkmotor im Stabilisator ermöglicht dabei das gezielte Aufbringen von Differenzkräften an einer Achse.



Bei der Entwicklung von GCC ist zu beachten, dass eine hohe Abhängigkeit zwischen Funktionalität, Softwarearchitektur und *Elektrik/Elektronik-Architektur (E/E-Architektur)* besteht. Ein ganzheitlicher Ansatz für die GCC-Wirkzusammenhänge ist übersichtlich in der Abb. 3.8 dargestellt. Die folgenden Abschnitte beschreiben nach [SR03] mögliche Lösungsansätze für die Realisierung des GCC-Fahrzeugs. Zur Beherrschung des entstehenden Komplexitätsgrades ist eine klare Funktions- und Gerätehierarchie mit definierten einheitlichen Schnittstellen notwendig.

### Ganzheitliche Regelungen

Für die ganzheitliche Regelung der Fahrdynamik existieren grundsätzlich zwei Ansatzmöglichkeiten:

1. Koexistenz mehrerer Regelungssysteme und
2. Integration der Regelungssysteme.

Am Beispiel der Horizontaldynamikregelung sollen die beiden Ansätze erläutert werden. In der Abb. 3.9 wird der Ansatz der Koexistenz der Dynamikregelungen für die Systeme Bremsen, Lenkung, Feder/Dämpfung und Antriebsstrang dargestellt. Die einzelnen Regler der Systeme müssen bezüglich ihres Wirkungsbereiches so abgestimmt werden, dass sie sich nicht negativ beeinflussen. Diese Reglerentkopplung führt zwangsweise zu einer Funktionseinschränkung. Des Weiteren benötigt jeder Regler eine eigene Fahrzustandserkennung mit Signalaufbereitung. Die Absicherung der Reglerwirkbereiche, der einzeln entwickelten und parametrisierten Systeme, ist mit einem großen Aufwand verbunden.

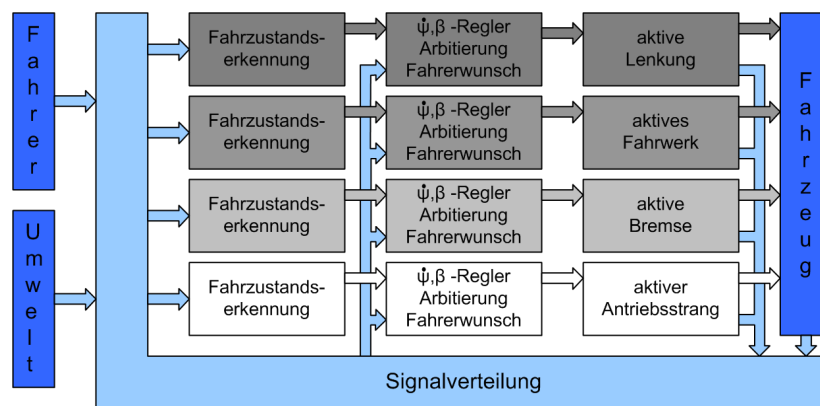


Abbildung 3.9: Horizontaldynamikregelung - Koexistenzansatz nach [SR03]

Nach [SR03] bietet daher die Integration der Regelungssysteme, siehe Abb. 3.10, eine bessere Lösung, dabei erfüllen die Einzelsysteme Lenkung, Bremse, Feder/Dämpfer und der Antrieb ihre Grundfunktionen unter Einbeziehung des Fahrerwunsches. Bezüglich der Horizontaldynamik bleiben diese Grundfunktionen auf eine reine Steuerung beschränkt. Über ein Bussystem sind die Einzelsysteme mit dem Gesamthorizontaldynamikregler verbunden und melden diesem die momentane Stellreserve und -dynamik. Die Parameter der Steuerungen werden bezüglich sich verändernder Fahrzeug- und Umweltbedingungen vom Gesamthorizontaldynamikregler adaptiert. In fahrkritischen Situationen berechnet der zentrale Horizontaldynamikregler unter Kenntnis des Fahrzustandes und Fahrerwunsches ein Korrekturmoment und verteilt es auf die einzelnen Aktuatoren Bremse, Lenkung, Fahrwerk und Antrieb.

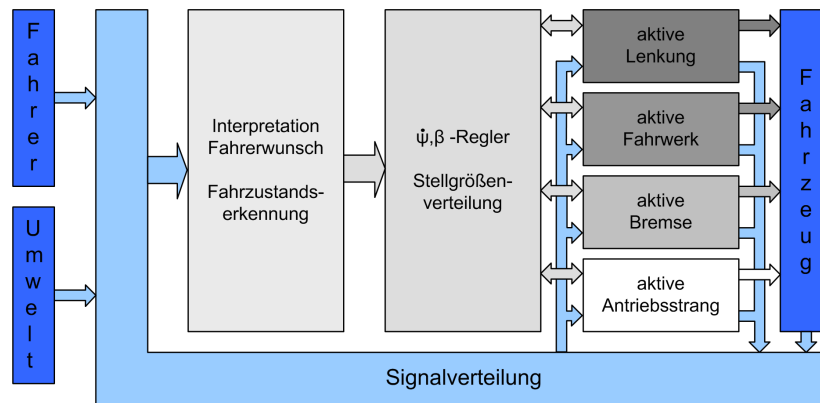


Abbildung 3.10: Horizontaldynamikregelung - Integrierter Ansatz nach [SR03]

### Softwarearchitektur

Zur Beherrschung der steigenden Software-Komplexität im vernetzten Chassis muss die Software zusammensetzbar sein und zu einem hohen Grad aus Standardmodulen bestehen. Das Gesamtsystem GCC muss in Form von Subsystemen mit schmalen Schnittstellen im Sinne einer objektorientierten Funktionsstruktur mit einem klaren Zeitverhalten angelegt werden. Die Einbettung der Software-Funktionen sollte dabei in der obersten Schicht des ISO/OSI-Schichtenmodells erfolgen.

### E/E-Architektur

Die E/E-Architektur umfasst die Gerätetopologie (Aktuatoren, Sensoren und Steuergeräte), die Busarchitektur und die Bordnetzarchitektur. Die intelligenten Module sind untereinander und mit dem Bedienungsmodul vernetzt. Selbstverständlich sind sicherheitsrelevante Systeme wie Lenkung und Bremse konsequent redundant auszulegen, damit die Ausfallsicherheit jederzeit gewährleistet ist. Der heute im Chassis eingesetzte ereignisgesteuerte CAN-Bus stößt dabei an seine Leistungsgrenzen. Für zukünftige vernetzte X-By-Wire-Anwendungen sind deshalb deterministische und fehlertolerante Bussysteme mit hohen Datenraten zu entwickeln.

### 3.3.4 Das X-By-Wire-Projekt

Mit dem europäischen Projekt X-By-Wire wurden die Grundlagen für zukünftige Fahrzeugkonzepte gelegt. Hauptziel dieses Projekts war es, Spezifikationen für die elektronische Datenübertragung im Automobil zu liefern [Bee01]. Die X-By-Wire-Systeme sind Bestandteil der Mechatronik und beschäftigen sich mit der elektrischen und elektronischen Energie- und Signalübertragung im Automobil. Im Folgenden soll kurz das X-By-Wire-Konzept beschrieben werden.

Unter den so genannten X-By-Wire-Systemen für Lenkung (Steer), Bremsen (Brake), Antriebsstrang und Fahrwerk werden im allgemeinen Fahrzeugsysteme verstanden, bei denen Erfassung und Umsetzung des Fahrerwunsches mechanisch/hydraulisch entkoppelt sind. Die Energie-/Signalübertragung erfolgt elektrisch/elektronisch. Als „reine“ oder „trockene“ X-By-Wire-Systeme werden Systeme verstanden, die keine mechanische/hydraulische Rückfallebene besitzen. Die Vorteile dieser Systeme sind:

- Neue konstruktive Freiheiten und damit Erhöhung der passiven Sicherheit,
- Reduzierung der Kosten durch neue mechatronische Komponenten,
- Verbesserung des Fahrkomforts und
- Erhöhung der aktiven Sicherheit durch ganzheitliche Regelungen.

X-By-Wire-Systeme müssen jedoch selbst im Falle eines fatalen Fehlers (Software- sowie Gerätefehler) die vorgeschriebene Funktionalität in jedem Fall erfüllen. Aus Sicht der Fahrzeugelektronik sind dazu gänzlich neue Konzepte notwendig. Die prinzipielle Auslegung eines X-By-Wire-Systems erfolgt als verteiltes, fehlertolerantes System. Wobei die Verlässlichkeit der geforderten Gesamtfunktionalität jederzeit garantiert werden muss.

### 3.3.5 Ausblick

Die Möglichkeiten, Fahrzeugfunktionen durch den Einsatz von Elektronik und Software zu verbessern, sind bei Weitem noch nicht ausgeschöpft. Die Zuwachsraten bezüglich Softwareumfang und Systemkomplexität steigen dabei jedoch exponentiell an [KG00]. Deshalb erfordern die gestiegenen Funktionen völlig neue Vorgehensweisen bei Entwicklung, Test und Optimierung. Besonders sind dabei mechatronische und simulationstechnische Probleme zu lösen.

Die Fahrzeugfunktionen für eine ganzheitliche Regelung des Automobils stehen heute bereits in Forschungsarbeiten zur Verfügung. Die Probleme, die dabei mit einem sicheren 42-V-Stromversorgungsnetz und entsprechenden Bustechniken für die Übertragung der gestiegenen Datenmengen bestehen, müssen noch gelöst werden.

## 3.4 Elektronik im Kraftfahrzeug

Der Einsatz von elektronischen Systemen stieg in den letzten Jahren rapide an. Immer mehr Hardware-Systeme (mechanisch/hydraulisch) werden durch elektronische Systeme und die darauf aufbauenden informationstechnischen Systeme – so genannte *Softwaresysteme* – ersetzt. Dabei hat die Informationstechnik in den letzten Jahren einen entscheidenden Einfluss

sowohl qualitativ als auch quantitativ auf die Elektronik genommen. Bedingt durch steigende Forderungen des Gesetzgebers sowie durch hohe Kundenerwartungen an Fahrkomfort, Agilität und Sicherheit nimmt die Komplexität der Steuergerätesoftware rasant zu.

Die Vernetzung der Steuergeräte ist seit den 90er Jahren Stand der Serienproduktion und ermöglicht die Realisierung völlig neuer Fahrzeugfunktionen, wie beispielsweise das ASR. Die Vernetzung von Steuergeräten im Fahrzeuginnenraum wird dabei genauso konsequent vorangetrieben wie die Vernetzung von Antriebs- und Fahrwerkssystemen. So kommen in der aktuellen S-Klasse von *DaimlerChrysler* in der Volllausstattung über 60 Steuergeräte zum Einsatz [Lam01].

In der Abb. 3.11 ist der prinzipielle Aufbau elektronischer Fahrzeugsysteme dargestellt. Ein Steuergerät wirkt über Aktuatoren mechanisch, hydraulisch oder elektrisch auf den Fahrzeugprozess ein und erhält über Sensoren Informationen über den aktuellen Fahrzustand [Lam01]. Die Signalübertragung zwischen Fahrzeugprozess und informationsverarbeitender Einheit erfolgt elektrisch. Funktionsabhängig erfolgt die Verbindung von Sensoren, Aktuatoren und Steuergeräten über analoge Signale, digitale Signale, serielle Datenleitungen oder Datenbusse.

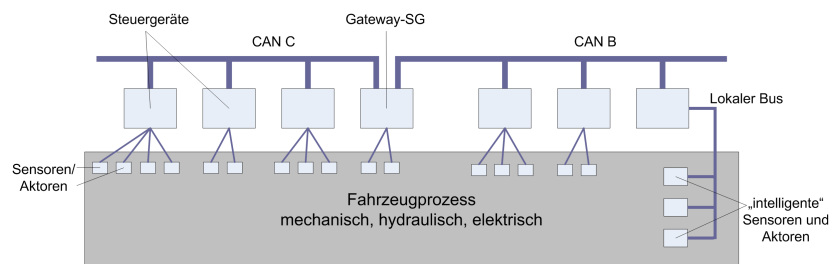


Abbildung 3.11: Struktur elektronischer Fahrzeugsysteme nach [Lam01]

Bussysteme werden hauptsächlich zur Vernetzung von Steuergeräten untereinander eingesetzt. Im Allgemeinen wird heute im Automobilbereich der *CAN-Bus* (*Controller Area Network-Bus*) eingesetzt, wobei funktionsabhängig unterschiedliche Klassifizierungen mit unterschiedlichen Übertragungsraten zum Einsatz kommen. So wurden beispielsweise im Antriebsstrang ein CAN Class C mit einer Datenübertragungsrate von 500 kbits/s und im Innenraum ein CAN Class B mit hoher Vernetzung eingesetzt. Bei in Zukunft weiter steigenden Datenmengen im Fahrwerks- und Antriebsbereich stößt der CAN-Bus jedoch an seine technologischen Grenzen. Die Entwicklung von neuen Bustechniken für den Automobilbereich ist daher notwendig.

Eine weitere Aufgabe der Fahrzeugelektronik stellt die Überwachung und Diagnose der Fahrzeugfunktionen dar. So sind in heutigen Steuergeräten umfangreiche Diagnosefunktio-

nen integriert. Sie überwachen den fehlerfreien Betrieb von Aktuatoren, Sensoren und Steuergeräten. Zum Beispiel betragen die Diagnosefunktionen in einem Motorsteuergerät ca. 20% der Gesamtfunktion.

Nach [Güh03] sind häufig erhebliche Softwarefehler in dem Programmcode der Steuergeräte enthalten, wodurch die Funktionsweise der Steuergeräte gestört wird. In keinem Fall darf dadurch eine fahrkritische Situation entstehen. Folglich ist der Softwaretest, in den ein Großteil des Gesamtaufwands eines Softwareprojekts investiert werden sollte, entscheidend. Da bei zukünftigen Entwicklungen die Systemsicherheit noch gesteigert werden muss, kommt dem Bereich der Fahrzeugdiagnose eine entscheidende Rolle zu. Durch umfangreiche Steuerungs-, Regelungs- und Diagnosefunktionen steigt der Speicherbedarf in Steuergeräten ständig. So benötigte ein Motorsteuergerät im Jahr 2000 bereits einen Speicherbedarf von 1 MByte gegenüber 32 kByte im Jahr 1990.

Wegen der gestiegenen Komplexität von Softwaresystemen ist die Anwendung neuer Methoden bei der Entwicklung und Bedatung der Software notwendig. In Abkehr von der früher ausschließlich eingesetzten Entwicklung und Prüfung im Fahrversuch wird heutzutage verstärkt die Simulation in der Entwicklung und Applikation eingesetzt [Güh03]. Die Testverfahren der Softwaresysteme sind dabei in den Entwicklungsprozess zu integrieren, siehe Abschnitt 3.5.1.

## 3.5 Simulation in der Automobilentwicklung

Der heutige Entwicklungsprozess eines Fahrzeugs von der Spezifikation über die Serienfertigung bis zur Modellpflege wird von Simulationen begleitet. Durch ihren Einsatz verbessern sich folgende Teilaspekte der Produktentwicklung [Hol01]:

- Verkürzung der Entwicklungszeit,
- Verbesserung der Produktqualität,
- Beherrschbarkeit zunehmender Komplexität von Baugruppen,
- Verwendung automatisierter Test-, Diagnose- und Optimierungsverfahren und
- Kosteneinsparung.

Gerade der Widerspruch zwischen Kosten und Zeit bei steigender Komplexität stellt bei der Entwicklung einschließlich Simulation eine große Herausforderung dar. Im Folgenden werden der Entwicklungsprozess von Steuergeräte-Software aufgezeigt und für diesen sinnvolle Simulationsmethoden erläutert.

### 3.5.1 Entwicklungsprozess

Die Komplexität eines Fahrzeugs ist in den letzten Jahren durch die zunehmende Anzahl an Funktionen sowie hohe Zuverlässigkeit, Sicherheit, Variantenvielfalt und Skalierbarkeit stark angestiegen, wodurch der Entwicklungsprozess schwer beherrschbar wird. In der Automobilindustrie lautet daher die Vorgehensweise [SZ03]:

Divide et Impera! – Teile und beherrsche!

In der Fahrzeugentwicklung führte dies zu einer funktionalen Teilung des Fahrzeugs in die Subsysteme Antrieb, Fahrwerk, Karosserie sowie Multi-Media. Die kurzen Entwicklungszeiten bedingen eine weitere Teilung in untergeordnete Subsysteme und Komponenten sowie eine parallele Bearbeitung der Entwicklungsaufgaben (engl. *Simultaneous Engineering*). Nach der Entwicklung und Prüfung der einzelnen Komponenten erfolgt eine stufenweise Integration bis zum fertigen Endprodukt. Aus den komplexen Systemen in der Fahrzeugtechnik ist ersichtlich, dass der Entwicklungsprozess nur mit einer klaren Arbeitsteilung einerseits und einer hervorragenden Zusammenarbeit über mehrere Fachdisziplinen hinweg andererseits erfolgen kann. Die Simulation wird für Entwicklung und Test von speziellen Fahrzeugfunktionen (beispielsweise das ABS), aber auch für Untersuchungen von komplexen Systemen (beispielsweise ein komplexes Fahrzeugsystem) eingesetzt. So ist es möglich, immer größere Bereiche des Prototypenbaus durch Simulationen zu ersetzen. Bei Fahrzeugherstellern und Lieferanten führt dies zu Multi-Projekt-Simulationen auf allen Systemebenen.

Zur Beherrschung des Entwicklungsprozesses sind entsprechende Prozessmodelle einzusetzen. Die Prozessmodelle beschreiben das zeitliche Vorgehen zur Herstellung des Produkts und decken sämtliche Entwicklungsschritte ab. Zur Beschreibung von Entwicklungsprozessen existieren unterschiedliche Prozessmodelle. Ein Großteil der Prozesse kommt aus der Softwaretechnik, wo bereits früher die rasant steigende Komplexität der Programme ein systematisches Vorgehen bei der Entwicklung erforderte. In der Fahrzeugindustrie ist das *V-Modell* weit verbreitet. In [Güh03, Lam01, KG00, SZ03] werden an die Automobilentwicklung angepasste V-Modelle beschrieben.

Anhand des, in der Abb. 3.12 dargestellten, V-Modells lässt sich der Einsatz der Simulation für die Entwicklung von Steuergeräte-Software aufzeigen. In Abhängigkeit der Entwicklungsschritte kommen dabei verschiedene Simulationsmethoden zum Tragen. Wesentliche Merkmale des V-Modells sind im linken Zweig die Arbeitsstrukturierung (Top-Down-Phase) und im rechten Zweig die Integration und der Test des Systems (Bottom-Up-Phase), wobei bei jedem Schritt eine Verifikation und zum Schluss eine Validierung erfolgt. Das V-Modell wird während der gesamten Entwicklung iterativ angewendet. Die einzelnen Integrationsstufen haben folgende Bedeutung:

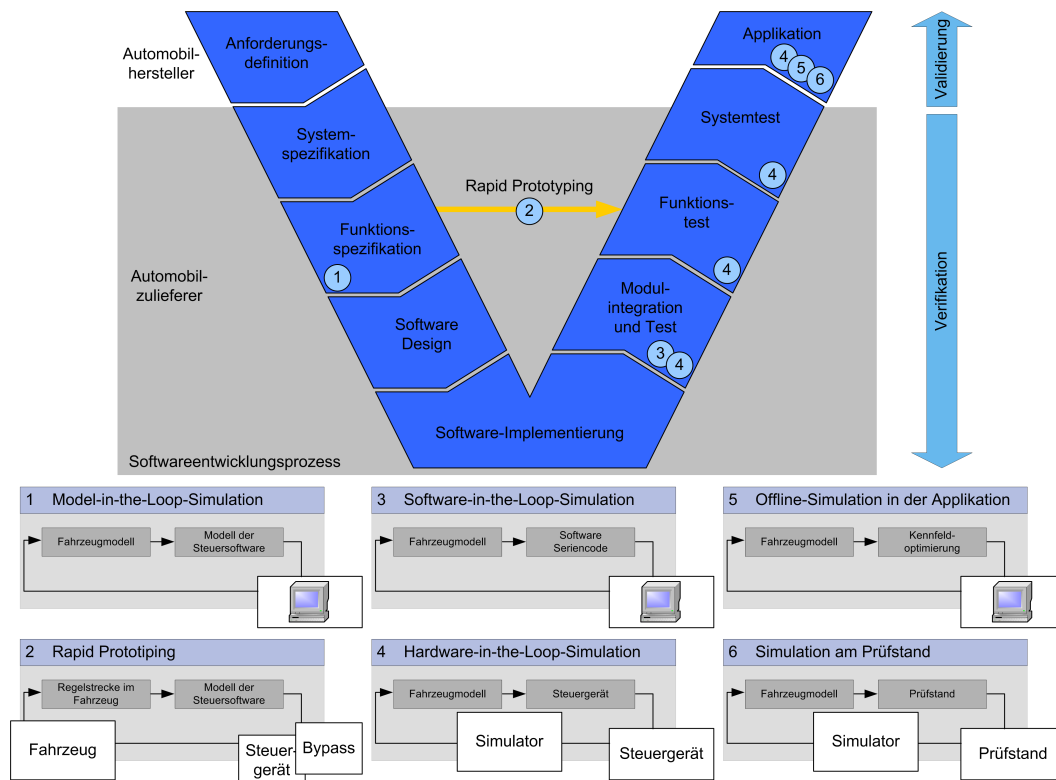


Abbildung 3.12: Software- und Funktionsentwicklungsprozess mit dem V-Modell [Güh03]

### • Anforderungsdefinition

Unter Berücksichtigung der Zielaufgabe des Kunden wird eine logische Systemarchitektur in einem Lastenheft festgelegt.

### • Systemspezifikation

Die logische Systemarchitektur wird analysiert und eine technische Systemarchitektur erarbeitet. Diese Analyse wird von den unterschiedlichen Fachdisziplinen getragen. Durch die virtuelle Produktentwicklung (siehe Abschnitt 3.5.3) kann die Systemspezifikation unterstützt werden.

### • Funktionsspezifikation

Die Funktionen werden grafisch in Softwaremodellen im Detail entwickelt und als Ergebnis in ein elektronisches Lastenheft abgelegt. Die Funktionsspezifikation kann durch eine *Model-in-the-Loop-Simulation* (MiL-Simulation) (siehe Abschnitt 3.5.2) unterstützt werden. Des Weiteren kann ein *Rapid-Prototyping-Prozess* (siehe Abschnitt 3.5.4) zur frühzeitigen Beseitigung von Spezifikationsfehlern angestrengt werden.

### • Software Design

In diesem Schritt wird die Software-Spezifikation festgelegt, dazu gehören der Modul-



und Programmentwurf.

- **Software-Implementierung**

Nach der arbeitsteiligen Entwicklung erfolgt die Integration der unterschiedlichen Software-Komponenten zu einem Softwaresystem. Anschließend wird der für das Zielsystem notwendige Seriencode generiert. Hier kommt vor allem der automatischen Codegenerierung eine bedeutende Rolle zu.

- **Modulintegration und Test**

Der erhaltene Seriencode wird bei einem Modultest überprüft, dies kann durch die *Software-in-the-Loop-Simulation (SiL-Simulation)* erfolgen (siehe Abschnitt 3.5.2). Es folgt die Integration auf der Modulebene. Dazu wird die Steuergeräte-Software auf dem eigentlichen Steuergerät implementiert. Beim anschließenden Test wird das ausführbare Programm (Seriencode) mit dem Betriebssystem und den verwendeten Hardwarekomponenten geprüft. Es entsteht ein Subsystem, das mit Hilfe der *Hardware-in-the-Loop-Simulation (HiL-Simulation)* verifiziert werden kann (siehe Abschnitt 3.5.2).

- **Funktionstest**

Die entwickelten und getesteten Module werden zu einer Fahrzeugfunktion zusammengesetzt und mit der Zielfunktion verglichen. Beim Funktionstest des erhaltenen Subsystems ist die geforderte Prüftiefe nur mit einer HiL-Simulation zu bewältigen.

- **Systemtest**

In diesem Schritt erfolgen Integration und Test auf Systemebene, d. h. es werden funktionsfähige und verifizierte Subsysteme zu einem Gesamtsystem integriert. Hierbei ist ein ausreichender Systemtest ebenfalls nur mit Hilfe der HiL-Simulation zu bewältigen. Des Weiteren werden einzelne Funktionen parametrisiert. Das Steuergerät wird wiederum mit den Sensoren und Aktuatoren verbunden und getestet.

- **Applikation und Systemabnahme**

Nach der Applikation der gesamten Funktionalität folgen schließlich ein Systemtest der logischen Systemarchitektur sowie ein Akzeptanztest gegenüber den Benutzeranforderungen. Bei der Applikation und Systemabnahme können die HiL-, Offline- und Prüfstandssimulationen eingesetzt werden.

Für eine erfolgreiche Entwicklung ist das Testen des Systems und der Komponenten während des gesamten Entwicklungsprozesses notwendig. Jedoch werden dabei immer wieder folgende drei Fehler beobachtet [SRG03]:

- Vielfach sind ungenügende Spezifikationen für die gewünschte Funktionalität verfügbar.

- Infolge dessen fehlt zudem eine umfassende Spezifikation des Verhaltens der Steuergeräte im Fehlerfall. Da zu einem „Gut-Fall“ oft mehrere „Fehlerfälle“ gehören, entsteht eine Spezifikationslücke.
- Tests sind nicht in den Entwicklungsprozess integriert.

Diese Probleme werden noch verstärkt, weil Systeme teilweise von mehreren Zulieferern stammen. Ein funktionierender Entwicklungsprozess beim Zulieferer bedingt keine Abstimmung des gesamten Fahrzeugentwicklungsprozesses. Es besteht daher ein hohes Risikopotential, das nur mittels ausführlichem und in den Entwicklungsprozess integrierten Testen beherrschbar ist. Hierbei ist es sinnvoll, die Tests schon bei der Systemspezifikation zu definieren. Eine gute Zusammenarbeit zwischen Test- und Entwicklungsingenieur ist dabei unverzichtbar.

### 3.5.2 Simulations-Methoden

#### Modell-in-the-Loop-Simulation

Die MiL-Simulation stellt die herkömmliche Simulation dar, bei der Regelungs- oder Steuerungsfunktion sowie die Strecke des Systems liegen als Simulationsmodell vorliegen, beispielsweise als *Matlab/Simulink*- oder *Modelica*-Modell. Die Simulation erfolgt auf einem PC oder einer Workstation. Vorteile der MiL-Simulation sind:

- Analyse des dynamischen Systemverhaltens (open/closed loop),
- Reglerentwurf mittels klassischer Verfahren und
- Stabilitäts- und Robustheitsanalyse.

#### Software-in-the-Loop-Simulation

Bei der SiL-Simulation wird die zu entwickelnde Reglerfunktion in ausführbaren Seriencodes auf einem Zielrechner (PC/Workstation) simuliert, das restliche Modell liegt wie bei der MiL-Simulation als Simulationsmodell vor. Mit der SiL-Simulation können somit einzelne Funktionen des Seriencodes ohne die entsprechende Zielhardware getestet werden.

#### Hardware-in-the-Loop-Simulation

Bei der HiL-Simulation wird das Verhalten eines Fahrzeugs unter Einsatz von Software- und Hardware-Modellen simuliert. Dazu werden reale Bauteile des Fahrzeugs (Realteile) über ihre elektrische Schnittstelle mit dem Simulationssystem verbunden, welches das Verhalten der Realteilmgebung in Echtzeit nachbildet [Lam01]. Für den Test und die Optimierung von Steuergeräten bedeutet dies, dass die Testumgebung für das elektronische Steuergerät eine

Verhaltenssimulation der realen Umgebung leisten muss. Die Umgebung des Steuergerätes, d. h. Fahrzeug mit Fahrer und Umwelt, ist als Echtzeitmodell nachzubilden. In der Abb. 3.13 ist das Prinzip der HiL-Simulation dargestellt.

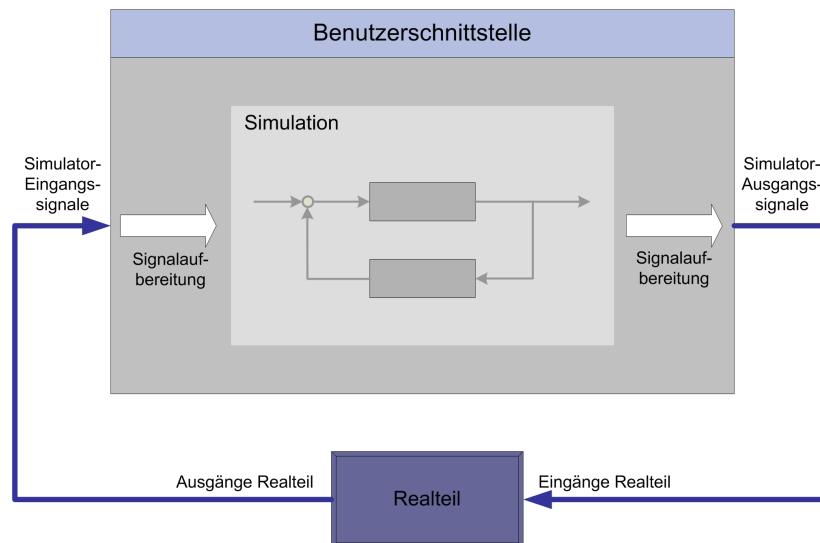


Abbildung 3.13: Prinzip der HiL Simulation nach [Lam01]

Im Zusammenhang mit der HiL-Simulation wird auch der Begriff des *HiL-Testsystems* gebraucht. Bei dem HiL-Testsystem handelt es sich um ein technisches System, das Anschluss, Betrieb und Test elektronischer Fahrzeuggeräte mit Hilfe der HiL-Simulation erlaubt. Neben einem echtzeitfähigen Simulationsmodell können weitere Modellkomponenten, wie beispielsweise Aktuatoren, aus realen Modellen bestehen. Aus Sicht des zu testenden Objekts darf sich die Testumgebung nicht von der Realität unterscheiden. Es besteht daher für das HiL-Testsystem die Forderung nach Vollständigkeit, Echtzeit und elektrische Unmittelbarkeit.

Der Einsatz der HiL-Simulation ist in der Steuergeräte-Entwicklung Stand der Technik. Sie stellt eine umfassende, kostengünstige, wiederholbare und automatisierbare Möglichkeit des Testens und Optimierens dar [Lam01]. Dabei werden sowohl durch den Test eines einzelnen Steuergeräts als auch des Steuergeräteverbunds, also des gesamten Elektronikersystems, in der frühen Entwicklungsphase eine höhere Entwicklungsreife und eine verbesserte Qualität im Serieneinsatz angestrebt. Die Entwicklung von Testsystemen ist aus Sicht der Vollständigkeit und Echtzeit für die Simulationstechnik nicht trivial. Außerdem sollte das verwendete Modell auch in anderen Entwicklungsschritten zur Verifizierung und Arbeitszeitverringerung einsetzbar sein.

### 3.5.3 Virtuelle Produktentwicklung

Die Erstellung eines virtuellen Fahrzeug-Prototyps auf einem Rechnersystem wird als virtuelle Produktentwicklung bezeichnet [Hol01]. Der virtuelle Prototyp wird dabei oft als *Digital Mock-Up (DMU)* bezeichnet. Am DMU können unterschiedliche Untersuchungen durchgeführt werden. In den letzten Jahren haben sich die Anwendungsfelder der DMU mit Simulation in der Entwicklung stetig ausgedehnt. Schwerpunkte der virtuellen Produktentwicklung sind Szenario-, Crash-, Fahrverhaltens- und Komfort-Simulationen.

Bei der Szenario-Simulation wird fotorealistisch ein Modell des Produkts (Fahrzeug) dargestellt, um die Marktchancen zu bestimmen. Mit Hilfe von Finite-Elemente-Modellen werden Crash-Simulationen zur Optimierung der passiven Sicherheit des Fahrzeugs durchgeführt. Einen weiteren wichtigen Punkt bildet die Beurteilung des Fahrverhaltens virtueller Prototypen. Das Ziel dabei ist es, die Auslegung des Fahrwerks anhand von Simulationen so zu optimieren, dass ein möglichst sicheres Fahrverhalten bis in fahrdynamische Grenzbereiche gewährleistet wird [Hol01]. Für diese Untersuchungen werden normalerweise sehr komplexe, nicht echtzeitfähige MKS-Modelle eingesetzt. Das charakteristische Schwingungsverhalten eines virtuellen Prototyps wird zur Komfortanalyse beispielsweise bei Motoranregung, Straßenanregung oder Reifenunwucht untersucht. Des Weiteren existieren Modelle zur Untersuchung des Geräuschverhaltens von virtuellen Prototypen.

Wichtiger Forschungsgegenstand in der virtuellen Produktentwicklung ist es, die Simulationswerkzeuge unterschiedlicher Fachdisziplinen zu verbinden, um so eine einheitliche Simulation zu erreichen. Einen möglichen Ansatz bietet die Modellierungssprache Modelica, siehe Kapitel 5.

### 3.5.4 Rapid Prototyping

Die rechnergestützte Entwicklung von Software für bereits bestehende oder ebenfalls in der Entwicklung befindliche elektronische Komponenten wird als Rapid Prototyping bezeichnet und stellt den nächsten Schritt nach der virtuellen Produktentwicklung dar. Gegenüber der virtuellen Produktentwicklung steht die Simulation einzelner Teilfunktionen im Vordergrund. Als Anwendungsbeispiele für den Einsatz von Rapid-Prototyping-Techniken sind z.B. die Entwicklung und Applikation von Motorsteuergeräten, Diagnoseverfahren, Fahrwerksregelsystemen und Fahrdynamikregelsystemen zu nennen [Hol01].

Beim Rapid-Prototyping-Prozess wird zuerst ein Modell erstellt, welches die Funktionsanforderungen aus der Systemspezifikation erfüllt. Dafür werden heute vorrangig grafische Simulationswerkzeuge eingesetzt. Zum Test und zur Optimierung der Funktionalität der Komponente wird eine SiL-Simulation angestrengt. Dabei wird die Modellkomponente mit weiteren

Modellen zu einem geschlossenen System verschaltet. Für die SiL-Simulation ist die Erstellung der Umgebung (z. B. Fahrer – Fahrzeug – Umwelt) entscheidend und nicht trivial. Hier bestehen Bemühungen, mit Hilfe der SiL-Simulation standardisierte, objektorientierte und multidisziplinäre Modelle der Umgebung zur Verfügung zu stellen. Es wird dabei besonderer Wert auf eine hohe Wiederverwendbarkeit der Komponenten gelegt.

Nach erfolgreicher Verifizierung bestehen grundsätzlich zwei Möglichkeiten, die Funktion der Komponente zu testen. Zum einen wird die Software-Komponente auf einem realen Steuergerät implementiert (Hardware) und in einer echtzeitfähigen Softwareumgebung getestet und optimiert. Dieses Testsystem wird allgemein als HiL-Simulation bezeichnet. Zum andern besteht die Möglichkeit, durch automatische Codegenerierung echtzeitfähigen Programmcode der zu untersuchenden Komponente (z. B. Getriebesteuergerät) zu erzeugen. Dieser Programmcode kann dann auf einer prototypischen echtzeitfähigen Plattform (z. B. Power-PC oder Signalprozessorsystem) implementiert werden (Software). Zum Test und zur Optimierung der Software-Komponente wird diese in einem bestehenden Prüfstand oder durch reale Fahrversuche untersucht. Dies stellt eine Sonderform der HiL-Simulation dar, wobei die Steuergerätekomponente in Software vorliegt. [Hol01]

### 3.5.5 Simulationswerkzeuge

In diesem Abschnitt soll auf die Simulationswerkzeuge, insbesondere auf ihre einzelnen Aufgabengebiete und ihre Bedeutung für die Fahrzeugindustrie, näher eingegangen werden. Dabei besteht kein Anspruch auf Vollständigkeit. Bei Simulationswerkzeugen ist eine grafische Oberfläche mittlerweile ebenso Stand der Technik wie die Möglichkeit Modelle, hierarchisch und modular zu strukturieren.

#### Finite-Elemente-Simulationswerkzeuge (FEM)

Finite-Elemente-Modelle beschreiben Bauteile als ein Netzwerk aus vielen kleinen, als elastisch angenommenen Körpern. Für die Modelle können ein-, zwei- oder dreidimensionale Elemente benutzt werden. Die Freiheitsgrade eines solchen Systems liegen über Hunderttausend [Hol01]. Diese Modelle werden unter harmonischer Anregung und der Annahme des eingeschwungenen Zustands im Frequenzbereich ausgewertet. Typische Beispielanwendungen sind Untersuchungen des Schwingungsverhaltens, des Crashverhaltens oder Aerodynamikanalysen. Dabei kommen Softwarewerkzeuge wie z. B. *ABAQUS*, *ANSYS* oder *LS-DYNA* zum Einsatz. Eine Echtzeitsimulation ist selbst mit leistungsfähiger Rechnerhardware wegen der Komplexität der FEM-Modelle nicht möglich.

### **Mehrkörper-Simulationswerkzeuge (MKS)**

Mehrkörpermodelle bestehen aus einer endlichen Anzahl von starren Körpern. Die Bindungen untereinander oder mit der Umwelt werden durch masselose Federn, Dämpfer oder Lager realisiert. Das mathematische Modell besteht im Allgemeinen aus einer Kombination von Bewegungsdifferentialgleichungen und arithmetischen Gleichungen. Systemabhängig kann das Modell aus mehr als hundert Freiheitsgraden bestehen. Innerhalb der virtuellen Produktentwicklung werden Mehrkörper-Simulationen zur Beschreibung von Längs-, Quer- und Vertikalbewegung des Gesamtfahrzeugmodells und der Teilmodelle eingesetzt. Die Softwaretools *ADAMS* und *SIMPACK* finden heute weite Verbreitung. In der Autoindustrie haben sich außerdem die Erweiterungen *ADAMS/CAR* und *SIMPACK Automotive+* etabliert.

### **Blockschaltbildorientierte Simulationswerkzeuge**

In der Steuerungs- und Regelungstechnik haben die blockschaltbildorientierten Simulationswerkzeuge weite Verbreitung gefunden. Besonders sei hier das Programm *Matlab/Simulink* genannt, das als Standard-Simulationswerkzeug betrachtet werden kann. Die Modellierung von Prozessen oder Funktionen auf einer systemtheoretischen Ebene mittels Wirkschaltplänen wird so ermöglicht. Die Modelle bestehen aus arithmetischen und logischen Operatoren, Variablen, Parametern, Ein- und Ausgängen und Subsystemen. Die Datenübertragung erfolgt im Allgemeinen durch Datenflusslinien und ist richtungsgebunden (Zuweisung). Aus der Anordnung und Verbindung von Operatoren und Subsystemen ergibt sich die Reihenfolge. In der Fahrzeugindustrie kommt für die blockorientierte Simulation auch das Simulationswerkzeug *ASCET-SD* zum Einsatz, das jedoch auf die Automobilindustrie beschränkt ist. Die Simulationswerkzeuge stellen außer einer grafischen Oberfläche auch automatische Codengenerierung zur Verfügung. Die blockschaltbildorientierten Modelle eignen sich durch die normalerweise ausgeprägte Idealisierung der Modelle hervorragend zu Echtzeitsimulation.

### **Objektorientierte Modellierung**

Für jede Fachdisziplin haben sich eigenständige Simulationswerkzeuge etabliert. Durch die immer komplexer werdenden Simulationen über mehrere Fachdisziplinen hinweg bestehen Bemühungen einer „echten“ multidisziplinären Modellierung, wobei ein Augenmerk auf die Echtzeitfähigkeit gelegt wird. Die Modellierungssprache *Modelica* verfolgt diese Zielsetzung. In dieser Arbeit wird in Kapitel 5 ausführlich auf die Möglichkeiten der objektorientierten Modellierung und der anschließenden Echtzeitsimulation eingegangen.

## 3.6 Zusammenfassung

In diesem Kapitel wurden die Möglichkeiten und Probleme der Fahrzeug-Regelungssysteme, Simulationssysteme sowie deren Stand der Technik aufgezeigt. Es wurde deutlich, dass nur durch entsprechende Simulationsmethoden neue und komplexe Regelungssysteme effizient entwickelt, getestet und optimiert werden können. Zur erfolgreichen Entwicklung von elektronischen Komponenten sind entsprechende Prozessmodelle und Simultaneous Engineering einzusetzen. Es wurde gezeigt, dass die Simulation nicht nur bei der Entwicklung von Fahrzeugkomponenten, sondern auch für den Test, die Optimierung und die Applikation einzusetzen ist. Des Weiteren ist darauf zu achten, den Testprozess in den Entwicklungsprozess zu integrieren.

Bei Voraussagen über das Systemverhalten durch Simulationen ist jedoch folgendes zu beachten:

- Qualität der erreichbaren Aussagen über das Systemverhalten – häufig sind dies Voraussagen – ist nur so gut wie das zugrunde liegende Modell [PS93], und
- Simulation darf nicht zum Selbstzweck werden.

Die Entwicklung von integrierten Regelungssystemen (Regelungssystemen im Verbund) erfordert eine ganzheitliche Betrachtung des Fahrzeugs. Somit ist für die Simulation das Streckenmodell bestehend aus Horizontal- und Vertikaldynamik zu modellieren. Die Echtzeitfähigkeit der Simulationsmodelle muss dabei für die Erstellung des HiL-Testsystems beachtet werden. Außerdem müssen für den Funktionstest der Systeme die erforderlichen Simulationsmodelle rechtzeitig zur Verfügung stehen. Es ist daher vorteilhaft, mit einheitlichen Modellen zu arbeiten, die während des gesamten Entwicklungsprozesses eingesetzt werden können. In dieser Arbeit wird deshalb der Ansatz der objektorientierten Modellierung mittels der Sprache *Modelica* zur Erstellung eines echtzeitfähigen Gesamtfahrzeugmodells untersucht.

## 4 Aufgabenstellung

Im vorherigen Kapitel wurde aufgezeigt, dass neue Innovationen im Automobilbereich nur mit Hilfe der Elektronik zu erreichen sind. Durch die einhergehende Mechatronisierung und Vernetzung der Teilkomponenten steigen die Anforderungen an die Modellierung und Simulation. Besonders im Bereich der Steuerungs- und Regelungstechnik wird der Bedarf an komplexen Modellen für die Steuergeräteentwicklung sichtbar. Zum Test und zur Optimierung von Steuerungen und Regelungen im kompletten Entwicklungsprozess steigt zudem der Bedarf an echtzeitfähigen Modellen. Im Besonderen sind detaillierte echtzeitfähige Gesamtfahrzeugmodelle interessant, die die Fahr- und Antriebsstrangdynamik berücksichtigen.

Im Rahmen dieser Arbeit wird ein echtzeitfähiges und objektorientiertes Modell des gesamten Fahrzeugs erstellt. Für die Simulation sind ebenfalls Fahrer und Umwelt zu berücksichtigen. Das Streckenmodell „Kraftfahrzeug“ steht im Mittelpunkt der Modellierung. Dabei baut das gesamte Konzept auf der objektorientierten Modellierungssprache *Modelica* auf. Zur Modellierung des Modells wird die von *Dynasim AB* entwickelte Simulationsumgebung *Dymola* eingesetzt. Zunächst wird eine kurze Einleitung in das Simulationssystem *Modelica* mit *Dymola* gegeben und das Simulationssystem mit dem Simulationsprogramm *Matlab/Simulink* verglichen.

Das Gesamtfahrzeugmodell baut auf den am Deutschen Zentrum für Luft- und Raumfahrt e. V. angefertigten Diplomarbeiten von Gerald Chimander [Cim03] und Stefan Heller [Hel03] auf. Dabei werden für die Modellierung die Bibliotheken „VehicleDynamics“ (frei) und „PowerTrain“ (kommerziell) verwendet. Aus den bestehenden Modellen für Antriebsstrang (Längsdynamikmodell) und Fahrdynamik (ohne Antriebsstrang) wird ein fiktives Gesamtfahrzeugmodell modelliert. Dazu werden die bestehenden Modelle eingehend untersucht. Das Gesamtfahrzeugmodell wird in die einzelnen Komponenten Fahrer, Antriebsstrang und Chassis entsprechend der Abb. 4.1 unterteilt und über entsprechende Schnittstellen miteinander gekoppelt.

Für den Antriebsstrang wird ein Modell aus der „PowerTrain“-Bibliothek, bestehend aus Motor und Automatikgetriebe mit Steuerungen, aufgebaut. Die Komponenten Differential, Bremsen, Gelenk- und Antriebswelle werden infolge zweckmäßiger Schnittstellenwahl dem Teilsystem Chassis zugeordnet. Für das Chassis wird aus der „VehicleDynamics“-Bibliothek



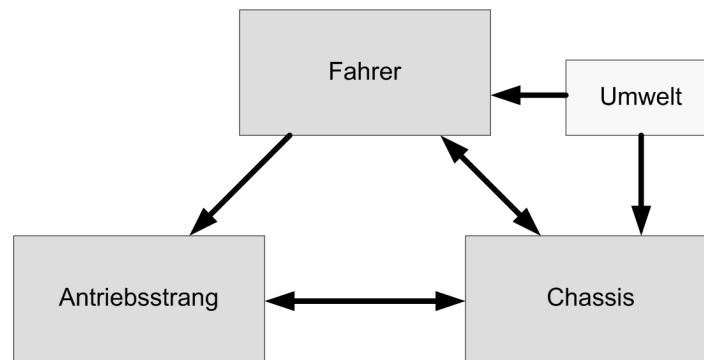


Abbildung 4.1: Struktur des Gesamtfahrzeugmodells

ein Fahrdynamikmodell für die Echtzeitsimulation angepasst. Zur Simulation verschiedener Fahrmanöver werden außerdem entsprechende Fahrermodelle entwickelt.

Bei der Modellerstellung ist es notwendig, möglichst wenige Idealisierungen durchzuführen, damit ein detailliertes Modell eines Fahrzeugs entsteht, das alle benötigten Daten zum Testen von Steuergeräten beinhaltet. Weiter soll das Modell die Steuergeräte für den Motor und das Automatikgetriebe enthalten. Das mit *Dymola* erstellte Gesamtfahrzeugmodell wird mittels *Dymola-Simulink-Interface* nach *Matlab/Simulink* importiert. Mit dem *Realtime-Workshop* wird dann echtzeitfähiger C-Code für das gesamte Fahrzeugmodell generiert und auf dem Echtzeitsimulationssystem *xPC Target* in einer echtzeitfähigen SiL-Simulation verifiziert. Für die Echtzeitfähigkeit soll eine Schrittweite von 1 ms eingehalten werden. Bei der Betrachtung wird besonders auf die Probleme bei der Echtzeitmodellierung eingegangen. Das echtzeitfähige Gesamtfahrzeugmodell wird mit den Arbeiten von [Cim03, Hel03] verifiziert. Zur Darstellung der Ergebnisse werden entsprechende Fahrzyklen simuliert und aussagekräftige Ergebnisse dargestellt. Außerdem wird ein detailliertes Fahrzeugmodell untersucht, das die dreidimensionalen Effekte des eindimensional modellierten Antriebsstrangs berücksichtigt. Zum Schluss wird das erstellte Fahrzeugmodell und die verwendeten *Modelica*-Bibliotheken für den Test und die Optimierung von Steuergeräten eingeordnet.

## 5 Objektorientierte Modellierung

In diesem Kapitel wird die objektorientierte Modellierung mit Hilfe der Modellierungssprache *Modelica* und der Simulationsumgebung *Dymola* beschrieben. Nach einer kurzen Einführung werden die Simulationsumgebung *Dymola* beschrieben und seine Vorteile aufgezeigt. Zum Schluss erfolgt ein Vergleich des Simulationssystems *Modelica* mit *Dymola* mit dem Simulationsprogramm *Matlab/Simulink*.

### 5.1 Einführung

Für die Simulation von detaillierten Fahrzeugmodellen werden Simulationenmethoden unterschiedlicher Fachdisziplinen (z. B. Mechanik, Elektrik und Hydraulik) benötigt. Bisherige Simulationssysteme wie beispielsweise *Matlab/Simulink* – für regelungstechnische Systeme, *ADAMS* – für Mehrkörpersysteme und *Flowmaster* – für hydraulische Systeme – berücksichtigen nur eine Fachdisziplin. Trotz umfangreicher Komponenten-Bibliotheken fehlen bestimmte Komponenten (besonders von anderen Fachdisziplinen). Die Modellierung von neuen Komponenten ist dabei nur eingeschränkt durch Aufruf externer C- oder Fortran-Prozeduren möglich.

Aus diesen Gründen wurde an einer „echten“ multidisziplinären Modellierung gearbeitet, die die Simulation von komplexen physikalischen Systemen ermöglicht. Hierfür bietet die objektorientierte Modellierung einen möglichen Ansatz. Unter objektorientierter Modellierung versteht man, dass Modelle komponentenweise so aufgebaut sind wie das reale System. Diese Methodik wurde 1978 von Hilding Elmqvist am *Lund Institute of Technology* (Lund, Schweden) entwickelt und seit Anfang der 90er Jahre vielerorts ausgebaut. Ab 1996 entwickelt die *Modelica Association* mit vielen Mitgliedern aus Industrie und Forschung die freie Modellierungssprache *Modelica*. Die Beschreibung von *Modelica*-Modellen erfolgt in ASCII-Notation, wobei die Beschreibung in Differentialgleichungen, algebraischen und diskreten Gleichungen deklarativ erfolgt, d. h. die Beschreibung ist gleichungsorientiert. Des Weiteren stellt die *Modelica*-Syntax Sprachkonstrukte für die grafische Darstellung bei der Modellierung sowie zur Gliederung und Dokumentation der Komponenten bereit.

Numerische Verfahren für die Simulation stellt *Modelica* jedoch nicht bereit. Daher kom-

men für die Simulation entsprechende Simulationsprogramme zum Einsatz. Das kommerzielle Simulationsprogramm *Dymola* bietet sehr gute Möglichkeiten zur grafischen Erstellung und Simulation von *Modelica*-Modellen. So können in der grafischen Oberfläche auf einfache Weise mittels Drag-and-Drop Modelle erzeugt werden. Selbstverständlich kann auch textbasiert direkt die *Modelica*-Syntax bearbeitet werden. Weiterhin ist *Dymola* der eigentliche Übersetzer der Modelle und stellt sehr gute Routinen zur Erstellung von simulierbarem Code zur Verfügung. Die Software *Dymola* ist sehr eng mit *Modelica* verzahnt, denn dessen Entwickler haben die Sprachentwicklung von *Modelica* wesentlich initiiert. Im allgemeinen Sprachgebrauch wird daher oft vom Simulationssystem *Dymola/Modelica* gesprochen. Es ist dabei zu beachten, dass *Modelica* auch mit anderen Simulationswerkzeugen [MODb] verwendet werden kann.

## 5.2 Modellierung mit Dymola und Modelica

Die Modellierung von *Modelica*-Modellen erfolgt weitestgehend grafisch in so genannten *Objektdiagrammen*. Die Objektdiagramme stellen eine Verallgemeinerung der Blockschaltbilder dar und ermöglichen eine einfache Sicht auf die Struktur des Modells. Typische Beispiele von Objektdiagrammen sind in der Abb. 5.1 dargestellt. Selbstverständlich können auch multidisziplinäre Modelle erstellt und simuliert werden.

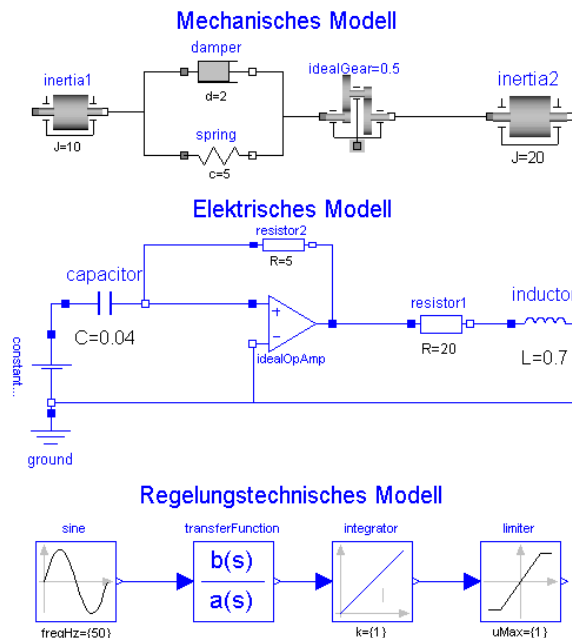


Abbildung 5.1: Drei Beispiele für *Modelica*-Objektdiagramme

Objektdiagramme sind aus verschiedenen Komponenten aufgebaut, wobei die Basiskom-

ponenten die niedrigste Ebene der Modelle darstellen. Jede Komponente entspricht dabei im Allgemeinen einer Komponente des realen Systems und wird unabhängig von der Umgebung definiert. Basiskomponenten (z. B. elektrischer Widerstand, Trägheit oder Integrator) werden durch Differentialgleichungen und algebraische Gleichungen (DAE) beschrieben. Zur Beschreibung werden dabei nur Variablen der Schnittstelle und lokale Variablen benutzt.

Weitere Komponenten bestehen aus verschalteten (Basis-)Komponenten und gegebenenfalls zusätzlichen Gleichungen. Die Verwendung der Komponenten basiert auf dem Prinzip der Instanziierung, d. h. Änderungen an einer Komponentendefinition wirken sich direkt in sämtlichen Instanzen der Komponente aus, die die geänderte Komponente beinhalten. Somit ist bei Beibehaltung der Schnittstellen ein einfacher Austausch der Komponenten möglich.

Komponenten werden über Schnittstellen (connector) miteinander verbunden. Die Verbindungen der Komponenten entsprechen dabei den physikalischen Verbindungen am realen System. In den Schnittstellen werden alle physikalischen Größen festgelegt, die über die Systemgrenze ausgetauscht werden. Im Allgemeinen sind die Größen ungerichtet, für Sonderformen (z. B. für die Regelungstechnik) existieren aber auch gerichtete Verbindungen. Es werden zwei Signaltypen unterschieden, die so genannten:

- „Potential-Variablen“ und
- „Fluss-Variablen“.

Potential-Variablen nehmen in den verbundenen Schnittstellen zwingend die gleichen Werte an. Beispiele dafür bilden der Flansch-Winkel oder die elektrische Spannung, siehe Abb. 5.2. Die Fluss-Variablen aller an einer Verbindung beteiligten Konnektoren summieren sich unter Beachtung der Vorzeichen zu Null (Erhaltungssatz), beispielsweise die Fluss-Variablen Moment, Kraft oder elektrischer Strom. Mit diesen Vereinbarungen kann das Simulationssystem *Dymola* aus mehreren verschalteten Komponenten auch ohne Kenntnis globaler Formalismen, wie z. B. den Kirchhoffschen Gesetzen, die korrekten Gleichungen des Gesamtsystems erstellen.

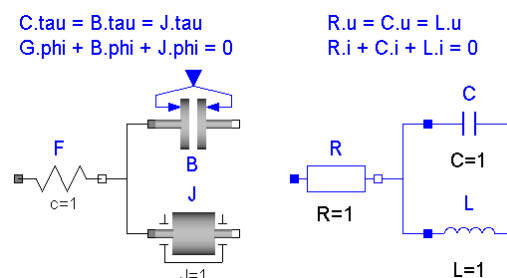


Abbildung 5.2: Verbindung von mechanischen und elektrischen Bauteilen in *Modelica*

Bei der Modellierung muss somit nicht schon bei der Eingabe nach einer Variable aufgelöst werden, sondern der *Modelica*-Übersetzer (*Dymola*) ist in der Lage, je nach Modellstruktur die Gleichungen passend aufzustellen. Das fertige Modell wird dazu durch den Übersetzer einer Signalfflussanalyse unterzogen und entsprechend umgeformt. Es entsteht ein großes, dünn besetztes DAE-System. Die Ein- und Ausgänge des Modells werden somit erst bei der Simulation festgelegt. Unter der Voraussetzung, dass alle Komponenten invertierbar sind, kann ein Modell durch einfaches Vertauschen der Ein- und Ausgänge leicht invertiert werden. Für die Simulation werden Variablen und Gleichungen des DAE-Systems weitestgehend durch symbolische Vorverarbeitung eliminiert. Weiter wird das System mittels effizienter Transformationen und einer Indexreduzierung in eine Zustandsform überführt. Für das meist deutlich verkleinerte System wird schließlich der für die Simulation benötigte C-Quellcode erzeugt. Mit diesen Verfahren ist es möglich, effizienten Programmcode zu erzeugen, der sich besonders für die Echtzeitsimulation eignet. Dem Anwender bleiben diese Umformungen jedoch weitestgehend verborgen. Die erstellte Modellstruktur bleibt zur Auswertung vollständig erhalten.

*Modelica* bietet darüber hinaus Sprachkonstrukte für die Modellierung von unstetigen und strukturvariablen Modellen (hybrid modelling) an. Bei der Simulation können entsprechende Simulatoren gezielt Ereignisse berücksichtigen. Die Simulation von diskret-kontinuierlichen Systemen wird ermöglicht mit If-then-else-Befehlen, boolsche Variablen sowie Operatoren für Zeitverzögerungen.

In *Modelica* werden die Komponenten hierarchisch in Bibliotheken verwaltet, die die Wissensbasis für die Modellierung darstellen. So existieren für viele Fachdisziplinen frei verfügbare Bibliotheken, wie beispielsweise:

- *Modelica* die Standard-Bibliothek mit den Teilbibliotheken für translatorische und rotatorische Mechanik, reglungstechnische Blockschaltbilder, SI-Einheiten, elektrische Grundelemente, eindimensionaler Thermik und mathematische Funktionen,
- *ModelicaAdditions* mit Teilbibliotheken für dreidimensionale Mehrkörpermechanik, Tabellen sowie Ergänzungen zu den Blockschaltbildern,
- *PowerTrain* enthält Komponenten des Fahrzeugantriebsstrangs (kommerziell),
- *VehicleDynamics* enthält Komponenten wie Fahrwerk, Reifen und Karosserie,
- *HyLib* für hydraulische Systeme (kommerziell) und
- *Thermofluid* enthält thermo-hydraulischen Komponenten.

Die Entwicklung von weiteren Bibliotheken wird ständig vorangetrieben. Auf der Internetseite von *Modelica* [MODa] werden weitere Bibliotheken vorgestellt. Selbstverständlich können

auf einfachste Weise grafisch oder in *Modelica*-Syntax eigene Bibliotheken mit neuen Komponenten erstellt werden. *Dymola* bietet hierzu einfache menügeführte Möglichkeiten, wobei auch ein einfacher Texteditor ausreichend ist.

Für die Modellierung und Simulation von *Modelica*-Modellen stellt *Dymola* zwei Betriebsmodi zur Verfügung:

- **Modellierungsmodus** um Modelle zu erstellen und zu bearbeiten (grafisch/textuell),
- **Simulationsmodus** zur Simulation des Modells und zur grafischen Darstellung der Ergebnisse in Zeitverläufen und gegebenenfalls 3D-Animationen.

Im Modellierungsmodus besteht die Möglichkeit, die Komponenten in vier verschiedenen Darstellungen zu bearbeiten. In der **Diagramm**-Darstellung werden die einzelnen Komponenten grafisch als so genannte *Icons* im Objektdiagramm dargestellt. Dies ermöglicht eine einfache Darstellung und Bearbeitung der Struktur des Modells. Jede Komponente kann mit dem Modell Browser geöffnet werden. So wird die Struktur der zugrunde liegenden Komponente sichtbar. Eine Bearbeitung ist wegen der objektorientierten Struktur selbstverständlich nur auf der obersten Ebene möglich. In der **Text**-Darstellung kann der Text des *Modelica*-Modells bearbeitet werden. In dieser Darstellung hat man die meisten Freiheitsgrade bei der Modellierung. Zum Modellverständnis muss jedoch die *Modelica*-Syntax beherrscht werden, was eine schnelle Einarbeitung nur bedingt ermöglicht. In der **Dokumentations**-Darstellung kann man sehr leicht die Dokumentation der Komponente einsehen und mittels *HTML* bearbeiten. Zum einfachen grafischen Erstellen des *Icons* einer Komponente dient die **Icon**-Darstellung.

Strukturelle Eigenschaften wie beispielsweise Komponenten, Schnittstellen und Verbindungen können auf einfachste Weise in der grafischen Oberfläche von *Dymola* bearbeitet werden, während Modifikationen von Gleichungen und Deklarationen im Texteditor stattfinden. Zudem unterstützt *Dymola* in der Diagramm-Darstellung menügeführt die Parametrierung des Modells. In der Abb. 5.3 ist der Modellierungsmodus in der Objektdiagrammansicht von *Dymola* dargestellt. Zur Erstellung und Bearbeitung der Komponenten sind der „Package-Browser“ und der „Component-Browser“ sehr hilfreich. Durch die verschiedenen Sichten auf Struktur, Gleichungen, einbezogene Quelltexte und Icons des Modells sind eine übersichtliche und effektive Modellentwicklung möglich.

In *Dymola* stehen für die Simulation verschiedene Integrationsalgorithmen wie beispielsweise Verfahren mit fester Schrittweite, z. B. „Euler“ und „Runge-Kutta“, oder Verfahren mit variabler Schrittweite, z. B. „DASSL“, zur Verfügung. Damit kann die Simulation optimal an die Bedürfnisse der Anwendung angepasst werden. Für die Grundparametrisierung und Verbesserung des Modells kann am Anfang des Entwicklungsprozesses eine „Offline-Simulation“

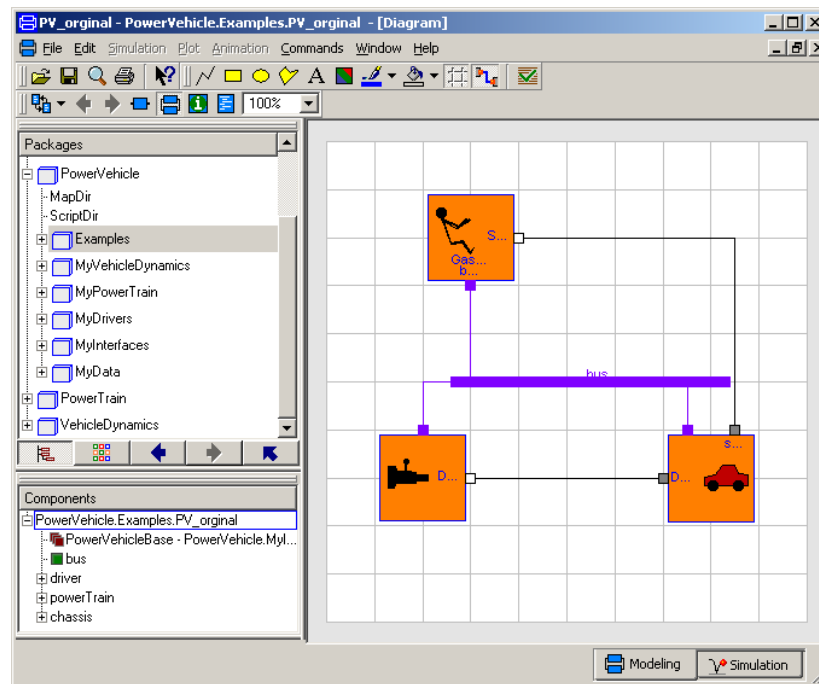


Abbildung 5.3: Bildschirmansicht von *Dymola*-Modellierungsmodus in der Diagrammansicht

eingesetzt werden, die keiner Echtzeitbedingung unterliegt. Somit kann ein Integrationsalgorithmus mit variabler Schrittweite eingesetzt werden. Diese Verfahren sind bei der Behandlung von strukturvariablen Systemen robuster und liefern durch angepasste Schrittweiten genauere Ergebnisse. Jedoch lassen sich damit keine Echtzeitsimulationen durchführen, weil die maximale Rechenzeit für eine bestimmte Schrittweite nicht vorgegeben werden kann. So kommen für die Echtzeitsimulation Verfahren mit fester Schrittweite zum Einsatz. Die entsprechenden Einstellungen für die Echtzeitsimulation des Gesamtfahrzeugmodells werden im Abschnitt 10.2 beschrieben.

## 5.3 Vergleich mit Matlab/Simulink

Bei heutigen Simulationsprogrammen ist eine grafische Oberfläche zur Modellierung sowie ein hierarchischer und modularer Aufbau der Modelle Standard. Selbstverständlich bieten die Programme *Matlab/Simulink* sowie *Modelica* mit *Dymola* diese Funktionalitäten. In diesem Abschnitt werden Vor- und Nachteile der beiden Simulationsprogramme aufgezeigt.

*Simulink* verwendet zur grafischen Darstellung die aus der Regelungstechnik bekannten Wirkungspläne (auch als Signalflußpläne bekannt). Der Vorteil von Wirkungsplänen ist, dass sie mathematische Gleichungen unmittelbar grafisch abbilden können. Aus der grafischen Struktur werden die zur Lösung des Systems benötigten differential-algebraischen Gleichungen

(DAE) erstellt. Dies kann systembedingt zu numerischen Problemen (algebraische Schleifen) führen, die nur iterativ gelöst werden können. Demgegenüber führt die objektorientierte Modellierung mit *Modelica* zu einer Modellstruktur (Objektdiagramme), die dem realen System gleicht. Wie im Abschnitt 5.2 beschrieben, besitzen *Modelica*-Modelle keine direkt verwendbare Struktur, sondern das Gleichungssystem wird vom *Modelica*-Übersetzer bestmöglich umgeformt und algebraische Schleifen werden gelöst. Außerdem kann der Modellierer, weil die Ein- und Ausgänge des Systems erst bei der Analyse festgelegt werden, meist auf einfache Weise durch Vertauschen der „Quellen“ und „Senken“ das Systemverhalten invertieren.

Bei beiden Programmen kann man auf eine Vielzahl von Komponenten aus Bibliotheken zurückgreifen, wobei es in *Modelica* einfacher ist neue Komponenten zu erstellen. Dabei muss der Modellierer sich um die mathematische Handhabung der Komponente in *Modelica*-Syntax kümmern. Viele Probleme wurden bereits in *Modelica*-Bibliotheken gelöst und müssen somit nicht erneut bearbeitet werden. In *Modelica* können des Weiteren Komponenten einfach durch neue Komponenten ausgetauscht werden.

In *Simulink* müssen alle anzuzeigenden Signale bei der Modellierung explizit durch separate Blöcke, z. B. Scopes, festgelegt werden. In *Dymola* hingegen stehen nach der Simulation alle Signale in der Modellstruktur zur Darstellung zur Verfügung. Jedoch kann es bei sehr komplexen Modellen aufwendig sein, die jeweiligen Signale zu beobachten, da es nicht möglich ist, entsprechende Plot-Konfigurationen direkt zu speichern. Außerdem ist die Bearbeitung von Diagrammen nur begrenzt möglich.

Über Scripte können in *Dymola* Simulations- und Ploteinstellungen gesteuert werden. An die Funktionalität von *Matlab*-Scripten reicht *Dymola* aber bei Weitem nicht heran. Auch bezüglich Auswertung, Analyse, Reglerentwicklung und zahlreichen weiteren in *Matlab* zur Verfügung stehenden Werkzeugen und Toolboxen für die verschiedenen Bereiche hat *Dymola* *Matlab* nichts entgegenzusetzen [RRS03]. *Dymola* bietet aber große Vorteile bei der Modellierung von multiphysikalischen Systemen.

Es ist sinnvoll, *Modelica* mit *Dymola* nicht als Konkurrenz zu *Matlab/Simulink* zu sehen, sondern die unterschiedlichen Stärken der beiden Programme zu nutzen. So speichert *Dymola* die Ergebnisse im von *Matlab* lesbaren *mat*-Format ab. Damit können die Ergebnisse mit *Matlab* weiter verarbeitet werden. Des Weiteren können mit dem *Dymola-Simulink-Interface* ganze Modelle aus *Dymola* in *Simulink* importiert und dort simuliert werden.



## 6 Antriebsstrangmodell

Im Folgenden werden die entsprechenden Antriebsstrangkomponenten des Gesamtfahrzeugmodells und ihre Funktionsweise beschrieben. Das Modell des Antriebsstrangs berücksichtigt aus Gründen der Rechenzeit und der unkomplizierten Modellerstellung nur eindimensionale Momente und Kräfte. Dazu werden für das Modellverständnis notwendige Komponenten erläutert. Auf Änderungen an Modellkomponenten gegenüber der „PowerTrain“-Bibliothek Version 1.0a wird an entsprechender Stelle eingegangen. Des Weiteren bietet selbstverständlich die Bibliothek weitere Komponenten für die Simulation von Antriebssträngen.

Am Ende dieses Kapitels wird ein Antriebsstrang mit der weiterentwickelten „PowerTrain“-Bibliothek Version 1.1 alpha 3 erstellt. Mit diesem Antriebsstrangmodell können die bei der eindimensionalen Modellierung vernachlässigten dreidimensionalen Effekte zusätzlich berechnet werden.

Grundsätzlich hat der Antriebsstrang die Aufgabe, das Fahrzeug durch Betätigung des Gaspedals zu beschleunigen. Dazu wird der Fahrerwunsch mittels Gaspedal ermittelt und ein Antriebsmoment an den angetriebenen Rädern erzeugt.

Einen Überblick über die oberste Struktur des mit *Dymola* modellierten Antriebsstrangs ist in der Abb. 6.1 dargestellt. Der Antriebsstrang enthält die Komponenten Verbrennungsmotor und Automatikgetriebe, die über einen Trilok-Drehmomentwandler mit Überbrückungskupplung verbunden sind. Des Weiteren existieren eine Busschnittstelle zur Datenkommunikation und ein mechanischer Flansch als Schnittstelle zum Fahrdynamikmodell. Im Folgenden werden wesentliche Komponenten des Antriebsstrangs beschrieben.

### 6.1 Motor

Der Verbrennungsmotor erzeugt durch Verbrennung eines Luft-Kraftstoff-Gemisches die für den Fahrzeugantrieb und den Betrieb der Nebenaggregate erforderliche mechanische Energie. Das Antriebsmoment wird dabei durch die Drosselklappenstellung bestimmt. Heutige Kraftfahrzeuge sind meist mit elektronischen Drosselklappenstellern (E-Gas) ausgeführt. Der Drosselklappenwinkel wird abhängig von der Gaspedalkraft, vom Motorzustand und gegebenenfalls weiterer Regeleinrichtungen wie ACC oder ESP durch das Motorsteuergerät (Motor-

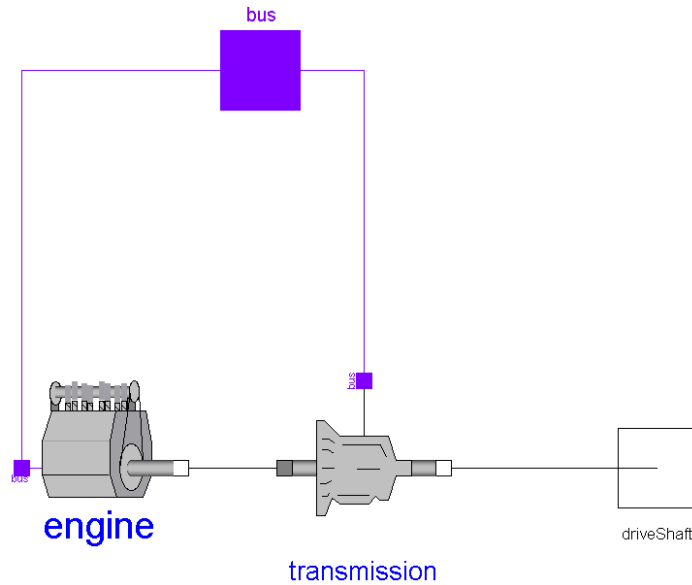


Abbildung 6.1: Objektdiagramm des Antriebsstrangs

management) geregelt.

In der Motorsteuerung wird die Drosselkappenstellung für die Berechnung des Motormoments berechnet. Das Modell berücksichtigt die Dynamik im Saugrohr und Motor nicht. Das Motormoment wird ohne die im realen Motor existierende Totzeit (ergibt sich aus der Verzögerung durch Ansaugung und Einspritzung bis zur Verbrennung) berechnet. Weitere Verbrennungsmotormodelle finden sich in den Arbeiten [Beu00, Mau03, Sch99].

In der Abb. 6.2 ist das *Modelica*-Modell des Motors dargestellt. Wesentliche Komponenten des Motormodells sind das Motor-Drehmomentmodell, die Motorsteuerung, der Anlasser und der Bus. Die Kurbelwelle wird als reine Trägheit betrachtet, d. h. ohne Elastizität. Das Motormoment  $M_M$  an der Schnittstelle **flange** ergibt sich aus den Momenten für Verbrennung  $M_{\text{mot}}$  und Anlasser  $M_{\text{anl}}$  zu:

$$M_M = M_{\text{mot}} + M_{\text{anl}} \quad . \quad (6.1)$$

### 6.1.1 Drehträgeit

Die Drehträgeit ist ein Basis-Element von *Modelica*, an ihr soll der Aufbau von *Modelica*-Komponenten erläutert werden.

Die Drehträgeit besteht aus den zwei Schnittstellen (Flansche) **flange\_a** und **flange\_b** sowie dem Trägheitsmoment  $J$ . Die Schnittstellen besitzen die Potentialvariable Drehwinkel  $\varphi$  und die Flussvariable Drehmoment  $M$ . Die Drehträgeit wird somit durch die drei Gleichungen vollständig beschrieben:

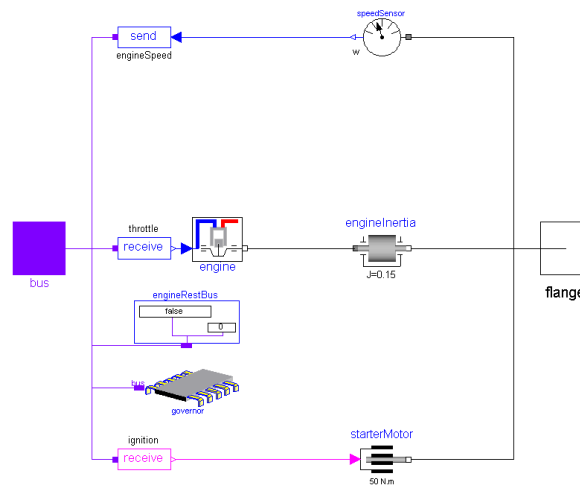


Abbildung 6.2: Objektdiagramm des Verbrennungsmotors

$$\omega = \dot{\varphi} \quad (6.2)$$

$$\alpha = \dot{\omega} \quad (6.3)$$

$$J\alpha = M_a + M_b \quad (6.4)$$

In *Modelica*-Syntax werden diese Gleichungen folgend geschrieben:

```
w = der(phi);
a = der(w);
J*a = flange_a.tau + flange_b.tau;
```

Der Ausdruck `der(phi)` stellt die Ableitung des Drehwinkels  $\varphi$  nach der Zeit dar. Die beiden Schnittstellen `flange_a` und `flange_b` sind eigenständige Komponenten und in die Trägheit mittels Vererbung eingebunden. Auf die Drehmomente `tau`, die eine Hierarchieebene tiefer liegen, wird mittels dem „Punkt-Operator“ (`flange_a.tau`) zugegriffen.

### 6.1.2 Motor-Drehmomentmodell

Für die Untersuchungen in dieser Arbeit sind die Drehmomentschwingungen des Verbrennungsmotors nicht von Bedeutung. Deshalb wird ein Mittelwertmodell zur Beschreibung des Motormoments herangezogen, welches in Hinblick auf die Rechenzeit optimal erscheint. In der Abb. 6.3 ist das Drehmomentmodell dargestellt. Die Berechnung des Motormoments  $M_{\text{mot}}$  basiert dabei auf einem Drehmoment-Kennfeld, das aus gemessenen statischen Stützpunkten eines realen Verbrennungsmotors besteht. In dem Drehmoment-Kennfeld sind die Verluste der

Nebenaggregate des Motors berücksichtigt. In Abhängigkeit der normierten Drosselklappenstellung  $\varphi_{DK}$  und der Kurbelwellendrehzahl  $\omega_{KW}$  wird ein Drehmoment an der Kurbelwelle berechnet:

$$M_{\text{mot}} = f(\varphi_{DK}, \omega_{KW}) \quad . \quad (6.5)$$

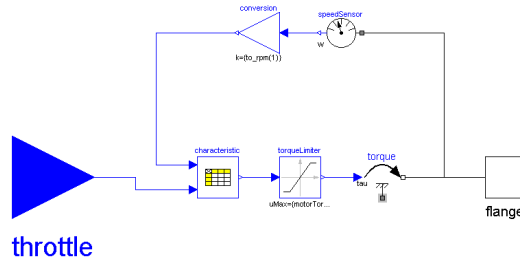


Abbildung 6.3: Objektdiagramm des Drehmomentmodells

### 6.1.3 Motorsteuerung

Die Motorsteuerung besteht aus einer Leerlaufregelung, einer Drehzahlbegrenzung und einer Drehmomentsteuerung für die Schaltphase, siehe Abb. 6.4. Die Leerlaufregelung und die Drehzahlbegrenzung sind in der Komponente „EC1“ realisiert und werden mit einem PI-Regler geregelt. Die Regelgröße ist die Kurbelwellendrehzahl  $\omega_{KW}$  und die Führungsgrößen sind die geforderte Drosselklappenstellung  $\varphi_F$  sowie die maximale  $\omega_{\max}$  und minimale  $\omega_{\min}$  Motordrehzahl, die nicht über- bzw. unterschritten werden dürfen. Die Stellgröße ist die normierte Drosselklappenstellung  $\varphi_{DB}$  und berechnet sich:

$$\varphi_{DB} = \begin{cases} \varphi_F(t) - (k \cdot \omega_{\text{emax}}(t) + \frac{k}{T} \int \omega_{\text{emax}}) & : \omega_{KW} \geq \omega_{\text{emax}} \\ \varphi_F & : \omega_{\text{emin}} < \omega_{KW} < \omega_{\text{emax}} \\ k \cdot \omega_{\text{emin}}(t) + \frac{k}{T} \int \omega_{\text{emin}} & : \omega_{KW} \leq \omega_{\text{emin}} \end{cases} \quad , \quad (6.6)$$

mit

$$\omega_{\text{emax}} = \omega_{\max} - \omega_{KW} \quad , \quad (6.7)$$

$$\omega_{\text{emin}} = \omega_{\min} - \omega_{KW} \quad . \quad (6.8)$$

Bei den Schaltphasen entstehen aufgrund der Drehzahlunterschiede vor und nach der Schaltung des Motors große Momente an den Rädern, die zu unerwünschtem Blockieren oder Durchdrehen der Räder führen können. Um diese zu verringern, wird in der Komponente „engineTorqueControl“ die Drosselklappenstellung  $\varphi_{DB}$  und damit das Drehmoment des Motors beeinflusst. Bei der Hochschaltung wird das Motordrehmoment für die Schaltphase begrenzt und bei einer Rückschaltung wird das Motordrehmoment angehoben. Dies wird über

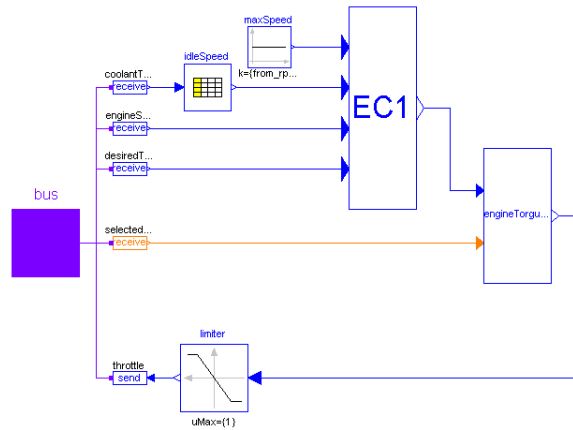


Abbildung 6.4: Objektdiagramm der Motorsteuerung

ein getriggertes Trapezsignal realisiert. Hierbei wird für die Hochschaltung eine Drosselklappenstellung  $\varphi_{DH}$  und für die Rückschaltung eine Drosselklappenstellung  $\varphi_{DR}$  vorgegeben. Somit berechnet sich die normierte Drosselklappenstellung  $\varphi_{DM}$  am Motor folgendermaßen:

$$\varphi_{DM} = \begin{cases} \min(\varphi_{DB}, \varphi_{DH}) & : \text{bei Hochschaltung} \\ \varphi_{DB} & : \text{sonst} \\ \max(\varphi_{DB}, \varphi_{DR}) & : \text{bei Rückschaltung} \end{cases} . \quad (6.9)$$

#### 6.1.4 Anlasser

Der Anlasser wird benötigt, um den Motor zu starten, d. h. für das Modell muss die Kurbelwellendrehzahl auf die Leerlaufdrehzahl angehoben werden. Der Verbrennungsmotor ist dabei fest über den Drehmomentwandler mit dem Automatikgetriebe verbunden. Beim Starten des Motors muss das Drehmoment  $M_{anl}$  des Anlassers, für die Zeit bis der Motor die Leerlaufdrehzahl erreicht hat, viel größer sein als die Summe aus dem Drehmoment  $M_{WP}$  an der Pumpenseite des Drehmomentwandlers und dem Drehmoment  $M_{mot}$  des Motors:

$$M_{anl} > M_{WP} + M_{mot} . \quad (6.10)$$

Das Drehmoment des Anlassers  $M_{anl}$  ist folgendermaßen definiert:

$$M_{anl} = \begin{cases} 150 \text{ Nm} & : 0 \leq t \leq 0.5 \\ 0 & : t > 0.5 \end{cases} . \quad (6.11)$$

#### 6.1.5 Bus

Für den Austausch von Informationen zwischen den einzelnen Steuergeräten, Sensoren und Aktuatoren werden heute im Automobil verschiedene Bussysteme eingesetzt. Demzufolge

wird im Modell auch ein Bussystem zum einfachen Austausch von Signalen zwischen den einzelnen Komponenten verwendet. Im Antriebsstrang hat sich der CAN-Bus etabliert. Er ermöglicht eine einfache und sichere Datenübertragung zwischen Motor- und Getriebesteuerung sowie Bremssystem. Der Bus im Modell ist dem CAN-Bus des Fahrzeugs nachempfunden, die speziellen CAN-Spezifikationen werden aber nicht berücksichtigt.

## 6.2 Automatikgetriebe

Erst durch das Getriebe ist ein effizienter Einsatz der Verbrennungsmotoren im Kraftfahrzeug möglich, denn Verbrennungsmotoren liefern nur in einem bestimmten Drehzahlbereich das für den Fahrzeugantrieb notwendige Drehmoment. Zur Anpassung an verschiedene Fahrzustände wie:

- Anfahren,
- Beschleunigen bei unterschiedlichen Steigungen und Geschwindigkeiten,
- Wirtschaftliches Fahren und
- Rückwärtsfahren

werden Getriebe unterschiedlichster Bauart eingesetzt. Für das Gesamtfahrzeugmodell wird ein angepasstes Automatikgetriebemodell aus der „PowerTrain“-Bibliothek verwendet. In der Abb. 6.5 ist das *Modelica*-Modell des Sechsgang-Automatikgetriebes dargestellt. Im Folgenden werden die wesentlichen Komponenten des Automatikgetriebes wie Drehmomentwandler, Getriebemodell, Gangwahlsteuerung und Steuerung der Kupplungsdrücke beschrieben.

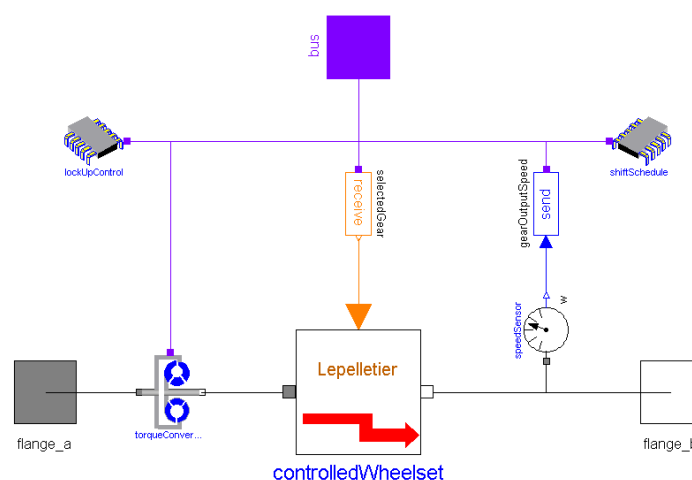


Abbildung 6.5: Objektdiagramm des Sechsgang-Automatikgetriebes

### 6.2.1 Drehmomentwandler

Der hydrodynamische Drehmomentwandler, im folgenden Wandler genannt, verbindet den Motor und das Automatikgetriebe miteinander. Gegenwärtig haben alle Kraftfahrzeugautomatikgetriebe den gleichen Wandleraufbau [DGK01]. Er besteht aus drei beschaufelten Rädern, dem Pumpenrad  $P$ , dem Turbinenrad  $T$  und dem Leitrad  $L$ , die in einem geschlossenen Kreislauf hintereinander geschaltet sind. Der Wandler ist mit Öl gefüllt und steht unter Druck. Die Kraftübertragung erfolgt über die durch die Schaufelräder erzeugte Ölströmung und gestattet so stufenlos variable Drehzahlverhältnisse. Damit ist das Anfahren und Schalten bei Automatikgetrieben ohne Kraftunterbrechung möglich.

Die Drehzahldifferenz zwischen Motor und Getriebe wird über den Wandler geschlossen. Bei großer Drehzahldifferenz, wie beispielsweise beim Anfahren, erfolgt über das still stehende Leitrad eine Drehmomentwandlung. Bei geringen Drehzahldifferenzen dreht das Leitrad lastfrei mit, hierbei erfolgt eine Drehzahlwandlung. Die Kraftübertragung kann durch diesen Aufbau nur bei einer Drehzahldifferenz erfolgen, weswegen hydrodynamische Wandler immer eine Verlustleistung von ca. 2-3% besitzen. Diese Verlustleistung kann mit einer Überbrückungskupplung verringert werden.

Das Modell des Wandlers und das Wandler-Kennlinienfeld sind in der Abb. 6.6 dargestellt. Der Trilok-Drehmomentwandler beruht auf gemessenen stationären Kennlinien, die das Übertragungsverhalten vollständig wiedergeben. Über dem Drehzahlverhältnis  $\nu = \frac{\omega_T}{\omega_P}$  sind die Momentverstärkung  $\mu_W$  und das am Pumpenrad anliegende Drehmoment  $M_{PC}$  dargestellt.

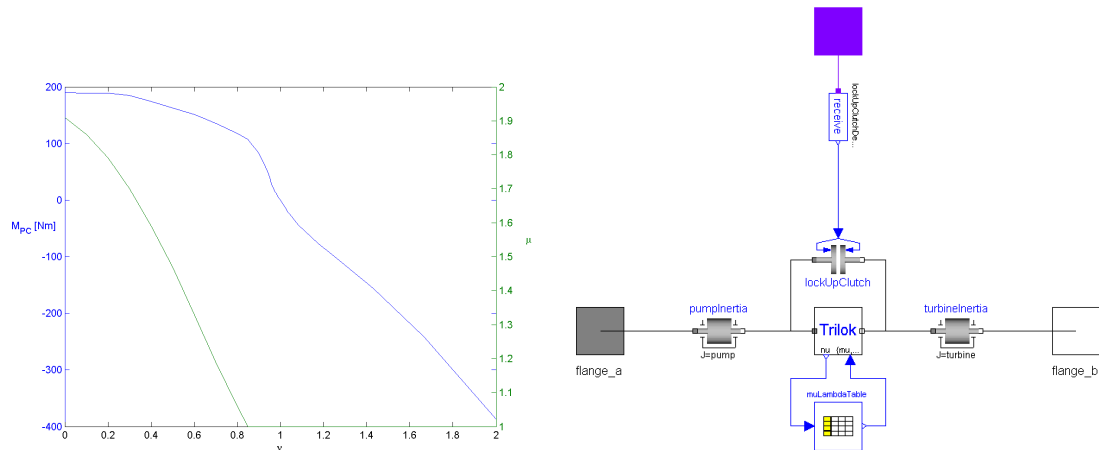


Abbildung 6.6: Wandlerkennlinien (links) und das Objektdiagramm (rechts) des hydrodynamischen Drehmomentwandlers

Für ein bestimmtes Drehzahlverhältnis  $\nu$ , der Pumpendrehzahl  $\omega_P$  und der Referenzdreh-

zahl  $\omega_{\text{ref}}$  kann so das Pumpenmoment  $M_P$  wie folgt berechnet werden:

$$M_P = M_{PC}(\nu) \left( \frac{\omega_P}{\omega_{\text{ref}}} \right)^2 . \quad (6.12)$$

Das Turbinenmoment  $M_T$  lässt sich mit Hilfe von Gleichung 6.12 und der Momentverstärkung  $\mu(\nu)$  berechnen:

$$M_T = \mu_W(\nu) M_P . \quad (6.13)$$

Des Weiteren sind die Trägheiten des Pumpen- und Turbinenrades und eine Überbrückungskupplung modelliert.

### 6.2.2 Elektronische Getriebesteuerung (EGS)

Die Aufgabe der EGS ist es, in Abhängigkeit verschiedener Einflussgrößen die Gangwahl des Getriebes zu realisieren. Dazu steuert oder regelt sie die entsprechenden hydraulischen Druckverläufe der Kupplungen und Bremsen im Getriebe. Weitere Aufgaben sind manuelle Eingriffe des Fahrers (z. B. direkte Gangwahl) auszuführen, Fehlbedienungen abzufangen, sowie die gesamte Ölversorgung und Kühlung des Getriebes sicherzustellen. Bei Automatikgetrieben mit Überbrückungskupplung wird außerdem die Überbrückungskupplung gesteuert. Das Modell der elektronisch-hydraulischen Getriebesteuerung gliedert sich in die drei Teilkomponenten:

- Gangwahl (Schaltprogramm),
- Brems- und Kupplungssteuerung und
- Überbrückungskupplungssteuerung.

Aus Gründen der Rechenzeit für das Echtzeitmodell wurde die Überbrückungskupplungssteuerung nicht modelliert. Somit findet im Modell keine Zuschaltung der Überbrückungskupplung statt.

#### Schaltprogramm

Im Schaltprogramm-Modell wird der zum Fahrzeugbetrieb notwendige Gang berechnet. Die entsprechenden Gänge werden gewählt in Abhängigkeit der Eingangssignale:

- Drehzahl der Getriebeausgangswelle,
- Drosselklappenwinkel,
- Programmwahlhebel und
- vom Fahrer geforderten Gang.



Der Fahrer betreibt das Automatikgetriebe im Allgemeinen im „Automatikprogramm“. In diesem Programm sind alle verfügbaren Vorwärtsgänge enthalten. Zentrale Elemente des Getriebebeschaltprogramms sind die Schaltkennlinien für Hoch- und Rückschaltung. Der einzulegende Gang wird abhängig von der Drosselklappenstellung  $\varphi_{DK}$  und der Getriebeausgangsdrehzahl  $\omega_{GA}$  berechnet. Die Auslegung dieser Schaltkennlinien geschieht unter Berücksichtigung der Leistungs- und Verbrauchsdaten des Motors, den Getriebeübersetzungen und der Fahrzeugdaten. Die Schaltkennlinien werden dabei nach optimalem Verbrauch einerseits und optimaler Leistungsausnutzung andererseits ausgelegt. Dabei ist auf geeignete Kennlinienwahl zu achten, so dass Pendelschaltungen vermieden werden.

Im Programm „Manuelle Gangwahl“ hat der Fahrer die Möglichkeit, den Gang direkt über einen Schalthebel oder per Tastendruck (Tipp-Funktion) zu wählen. In diesem Modus ist der Fahrer wie bei herkömmlichen Schaltgetrieben für die richtige Gangwahl verantwortlich, wobei Fehlbedienungen nicht zugelassen werden.

### Steuerung der Bremsen und Kupplungen

In Automatikgetrieben werden die Gänge kraftschlüssig geschaltet, d. h. durch die Betätigung von Reibkupplungen und -bremsen wird die Motordrehzahl automatisch dem neuen Gang angepasst, ohne dass die Verbindung zwischen Motor und Getriebe unterbrochen wird. In der Komponente Brems-/Kupplungssteuerung, siehe Abb. 6.7, werden aus dem Eingangssignal für den geforderten Gang die hydraulischen Druckverläufe der einzelnen Schaltelemente berechnet. Die Steuerung der Druckverläufe geschieht über ein trapezförmiges Signal. Die Dynamik des Drucksystems ist durch ein Verzögerungsglied 1. Ordnung approximiert.

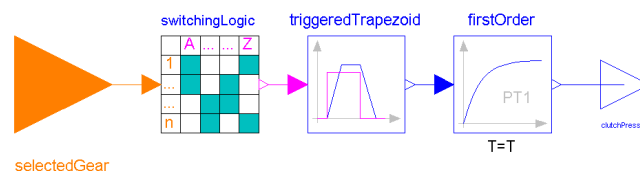


Abbildung 6.7: Objektdiagramm der Steuerung für Kupplungs- und Bremsdrücke

Über die Parameter:

- *maxPressure*: maximaler Öldruck in bar,
- *riseTime*: Anstiegsdauer der Druckrampe in s,
- *fallTime*: Abfalldauer der Druckrampe in s und
- *T*: Verzögerungszeit in s

können die verschiedenen Druckverläufe der Schaltelemente an den Gangwechsel angepasst werden.

### 6.2.3 Lepelletier-Radsatz

Zur Realisierung von verschiedenen Gangstufen werden Automatikgetriebe aus mehreren Planetensätzen zusammengesetzt. Die Vorteile der Planetengetriebe liegen in der Rotationssymmetrie, der hohen Leistungsdichte und insbesondere in der kompakten, coaxialen Bauweise [Haj02]. Das modellierte Lepelletier-Automatikgetriebe ist in der Abb. 6.8 dargestellt. Es wurde nach seinem Erfinder Lepelletier benannt und besteht aus einem Planetensatz, einem Ravigneaux-Planetensatz, zwei Lamellenbremsen (C und D) und drei Lamellenkupplungen (A, B und E).

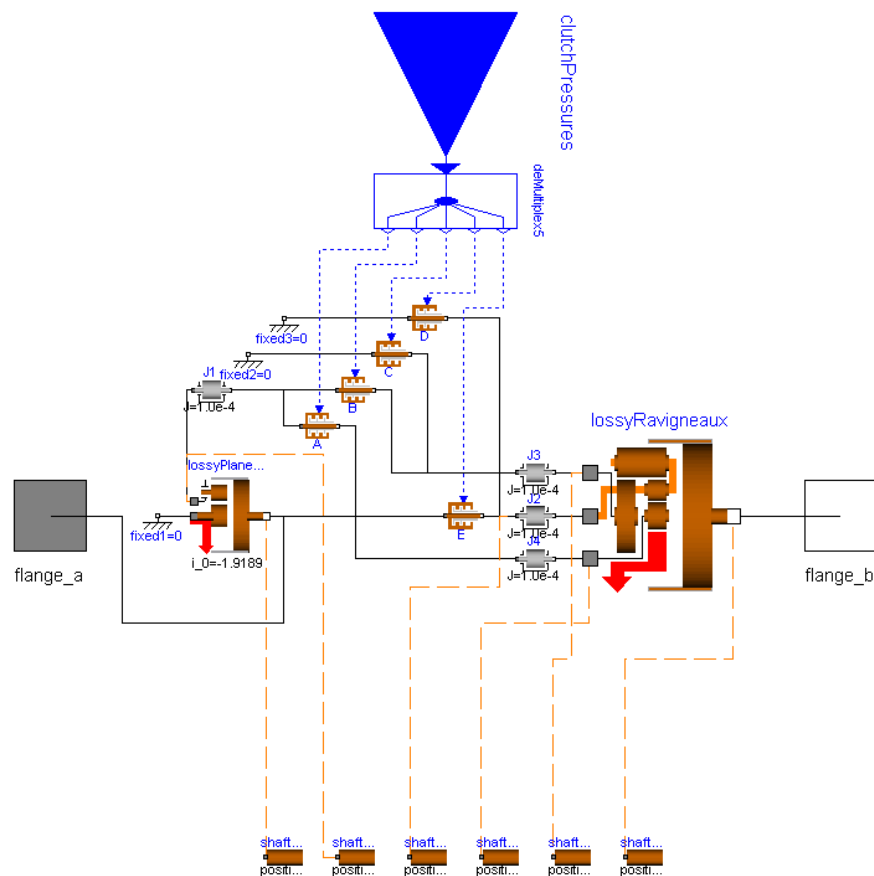


Abbildung 6.8: Objektdiagramm des 6-Gang-Lepelletier-Getriebemodells

#### Planetensatz

Ein Planetensatz besteht aus einem Sonnenrad (*sun*), einem Hohlrad (*ring*) und  $n$  identischen Planetenrädern. Die Planeten sind in einem Planetenträger (*carrier*) gelagert. Der Planeten-

träger, das Hohlrads und das Sonnenrad sind jeweils mit einer Welle verbunden. Die Übersetzungsverhältnisse sind durch die Anzahl der Zähne der jeweiligen Räder charakterisiert. Das Modell „LossyPlanetary“ beschreibt einen verlustbehafteten Planetensatz ohne Trägheiten. Die Besonderheit des Modells liegt in dem hybrid diskret-kontinuierlichen Modellierungsansatz der Reibung. Er ermöglicht eine robuste und effiziente Behandlung geschwindigkeits- und momentabhängiger Reibung und ist in [PSO66] ausführlich beschrieben. Das Reibmodell wird weiter unten in der Komponente Kupplung beschrieben.

Das durch Reibung entstehende Verlustmoment  $M_{\text{reib}}$  wird aus Kennlinien berechnet. Mit der statischen Getriebeübersetzung  $i_0$  ist das kinematische Verhältnis des Planetenradsatzes durch die Wilson-Gleichung definiert:

$$i_0 = \frac{\varphi_{\text{sun}} - \varphi_{\text{carrier}}}{\varphi_{\text{ring}} - \varphi_{\text{carrier}}} . \quad (6.14)$$

Der Drehmomentenausgleich ist folgendermaßen definiert:

$$-M_{\text{ring}} = i_0 (M_{\text{sun}} - M_{\text{reib}}) , \quad (6.15)$$

$$-M_{\text{carrier}} = (1 - i_0) M_{\text{sun}} + i_0 M_{\text{reib}} . \quad (6.16)$$

### Ravigneauxsatz

Der Ravigneauxsatz stellt eine spezielle Variante von Planetengetrieben dar, bei dem zwei Planetensätze zusammengefasst sind. Die beiden Planetensätze sind in einem Planetenträger gelagert und bei einem der beiden Planetensätze werden die Planetenräder so breit ausgeführt, dass sie anstelle des Hohlrads mit den Planeten des anderen Planetensatzes kämmen. In der Abb. 6.9 ist das animierte *Modelica*-Lepelletier-Getriebe dargestellt. Auf der Eingangsseite des Getriebes befindet sich der Planetensatz und auf der Ausgangsseite der Ravigneauxsatz. Dazwischen sind die entsprechenden Wellen und Kupplungen angeordnet. Die Planetensätze werden durch ein Planetenrad vertreten, der Planetenträger wird zur Veranschaulichung nicht angezeigt und mechanisch verbundene Komponenten sind in der gleichen Farbe dargestellt.

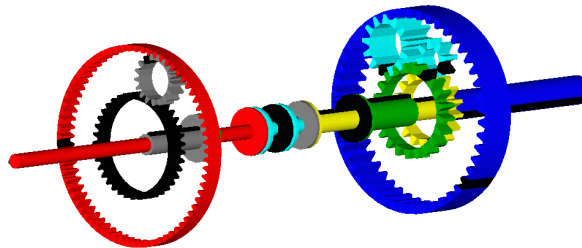


Abbildung 6.9: Animation des Sechsgang-Lepelletier-Getriebemodells

Das Modell „LossyRavigneaux“ beschreibt einen verlustbehafteten Ravigneauxsatz ohne Trägheiten. Das Verlustmoment wird entsprechend dem Planetengetriebemodell bestimmt.

Das kinematische Verhältnis ist mit den Getriebeübersetzungen zwischen dem kleinen Sonnenrad und dem Hohlrad  $i_{\text{ring}}$  und zwischen kleinem und großem Sonnenrad  $i_{\text{sun}}$  durch folgende Gleichung definiert:

$$\varphi_{\text{smallsun}} = i_{\text{ring}}\varphi_{\text{ring}} + (1 - i_{\text{ring}})\varphi_{\text{carrier}} \quad (6.17)$$

$$\varphi_{\text{smallsun}} = i_{\text{sun}}\varphi_{\text{bigsun}} + (1 - i_{\text{sun}})\varphi_{\text{bigsun}} \quad (6.18)$$

Der Drehmomentenausgleich ist folgend definiert:

$$0 = M_{\text{smallsun}} + M_{\text{bigsun}} + M_{\text{ring}} + M_{\text{carrier}} \quad (6.19)$$

$$0 = M_{\text{smallsun}} + \frac{M_{\text{ring}}}{i_{\text{ring}}} + \frac{M_{\text{bigsun}}}{i_{\text{sun}}} - M_{\text{reib}} \quad (6.20)$$

### Kupplung

Lamellenkupplungen und -bremsen werden als aktive Schaltelemente in Automatikgetrieben eingesetzt und stellen die wichtigsten Elemente des Gangwechsels dar. In diesem Modell besteht der Unterschied zwischen Kupplung und Bremsen allein in der Art der Verschaltung. Als Kupplungen werden die Elemente bezeichnet, die zwei drehbare Wellen miteinander verbinden. Bei Bremsen handelt es sich um Elemente, die eine drehbare Welle mit dem Getriebegehäuse verbinden. Die Lamellenkupplungen und -bremsen werden hydraulisch gesteuert und dienen zur Momentübertragung zwischen den Getriebebauteilen. Das im Lepelletier-Getriebe eingesetzte Kupplungsmodell basiert auf dem oben erwähnten Reibmodell.

Das Kupplungsmodell beruht auf dem hybriden Modellansatz, d. h. die Berechnungen für einen Zustand werden kontinuierlich durchgeführt, die Umschaltung zwischen den Zuständen erfolgt jedoch diskret. Im Modell wird zwischen geschlossener (**locked**), offener (**free**) und vorwärts/rückwärts (**forward/backward**) schleifender Kupplung unterschieden. Für jeden Zustand wird das Kupplungsmoment  $M_K$  gesondert berechnet. Die Erkennung des entsprechenden Reibzustandes wird sicher über einen Zustandsautomaten realisiert. Eine Umschaltung vom Haft- in den Gleitzustand erfolgt, wenn die äußeren Kräfte das maximale Haftreibungsmoment  $M_{\text{haft}}$  überschreiten. Umgekehrt wird vom Gleit- in den Haftzustand umgeschaltet, wenn  $\omega_{\text{rel}} = \omega_b - \omega_a = 0$  erfüllt ist. Die Umschaltung zwischen Gleitreibung und offener Kupplung erfolgt durch Auswertung der durch den Steuerdruck vorgegebenen Normalkraft  $F_N$ :

$$\mathbf{free} = \begin{cases} \text{true} & : F_N \leq 0 \\ \text{false} & : F_N > 0 \end{cases} \quad (6.21)$$

Die Normalkraft  $F_n$ , die auf die Reibflächen der Lamellenkupplung wirkt, wird in Abhängigkeit von der Federkraft  $F_f$  der Kolbenfläche  $A_{kb}$  und dem Hydraulikdruck  $p$  berechnet:

$$F_n = p \cdot A_{kb} - F_f \quad (6.22)$$

Über die Innen- und Außendurchmesser  $r_{in}$  und  $r_{au}$  sowie die Anzahl der Reibflächen  $n_A$  ist die Geometriekonstante  $c_{geo}$  definiert:

$$c_{geo} = n_A \frac{r_{au} + r_{in}}{2} . \quad (6.23)$$

Der Reibkoeffizient  $\mu_{gleit}$  während der Gleitphase ist abhängig von der Relativdrehzahl  $\omega_{rel}$  und dem Kupplungsdruck  $p$  und wird aus einem Kennfeld berechnet:

$$\mu_{gleit} = f(\omega_{rel}, p) . \quad (6.24)$$

Das maximale Haftreibungsmoment  $M_{haft}$  wird für  $\omega_{rel} = 0$  folgendermaßen berechnet:

$$M_{haft} = peak \cdot \mu_{gleit} \cdot c_{geo} \cdot F_N . \quad (6.25)$$

Hierbei stellt *peak* eine Konstante zur Verstärkung im Haftzustand dar. Für das Modell wird eine vereinfachte Reibwertkennlinie, wie in der Abb. 6.10 dargestellt, verwendet.

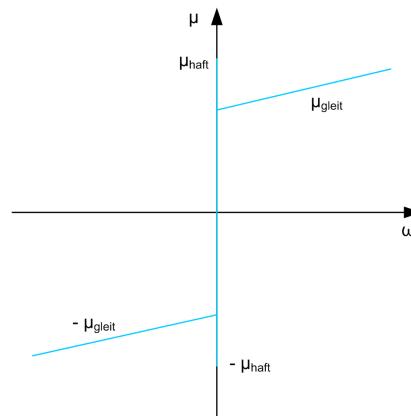


Abbildung 6.10: vereinfachtes Reibwertdiagramm

Das Kupplungsmoment  $M_K$  ergibt sich demzufolge zu:

$$M_K = \begin{cases} 0 & : \text{ free} \\ M_{flange\_b} = -M_{flange\_a} & : \text{ locked} \\ c_{geo} F_n \mu & : \text{ forward} \\ c_{geo} F_n (-\mu) & : \text{ backward} \end{cases} . \quad (6.26)$$

Hierbei stellen  $M_{flange\_a}$  und  $M_{flange\_b}$  die äußeren Drehmomente an den entsprechenden Schnittstellen `flange_a` und `flange_b` dar.

## 6.3 Antriebsstrang mit 3D-Effekten

Am Ende dieser Arbeit stand die inoffizielle Version 1.1 alpha 3 der „PowerTrain“-Bibliothek zur Verfügung. Sie beinhaltet Komponenten für die Berechnung der dreidimensionalen Effekte

(3D-Effekten) von eindimensionalen mechanischen Rotationssystemen. Auf der Grundlage des oben dargestellten Antriebsstrangmodells wurde damit ein Modell realisiert, das die 3D-Effekte in den einzelnen Komponenten berücksichtigt. Im Folgenden werden entsprechende Erweiterungen des Modells beschrieben und die Modellstruktur aufgezeigt.

In der Abb. 6.11 ist die Struktur des Antriebsstrangmodells mit den Verbindungen (dick schwarz) und Schnittstellen für die 3D-Effekte dargestellt. Die Verbindung zum Differentialgetriebe des Fahrdynamikmodells erfolgt weiterhin über die 1D-Schnittstelle (**driveShaft**) für mechanische Rotationssysteme. Zusätzlich werden jedoch die Abstütz- und Kreiselmomente von Motor und Getriebe berücksichtigt. Das Moment wird dabei räumlich in einem Punkt zusammengefasst und über die 3D-Schnittstelle (**frame\_a**) für die Fahrdynamikkomponente bereitgestellt.

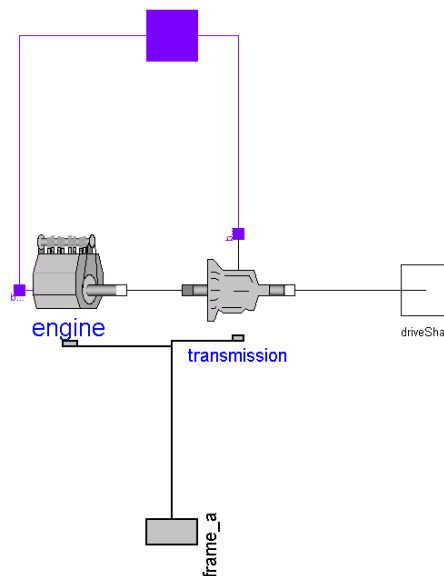


Abbildung 6.11: Objektdiagramm des Antriebsstrangmodells mit 3D-Effekten

### 6.3.1 Verbrennungsmotor

In der Abb. 6.12 ist die Struktur des Verbrennungsmotors dargestellt. Die Komponenten **engine** und **starterMotor** beinhalten eine zusätzliche Schnittstelle (**bearing**), die das eindimensionale Abstützmoment zum Gehäuse bereitstellt. In der Komponente **mounting** wird das eindimensionale Moment an einem 3D-Mehrkörpersystem befestigt. Des Weiteren wird das Kreiselmoment der Drehträgheit **engineInertia** für die Simulation berücksichtigt.

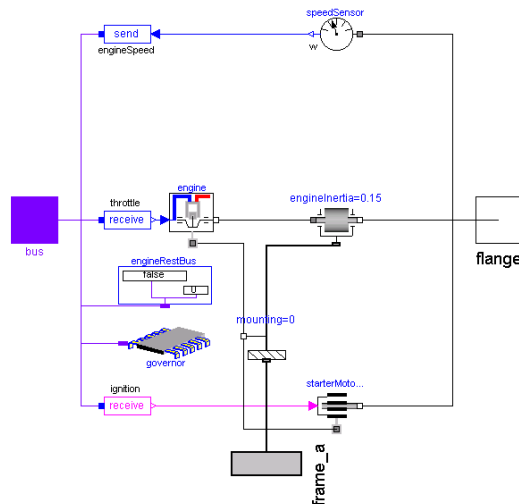


Abbildung 6.12: Objektdiagramm des Verbrennungsmotormodells mit 3D-Effekten

### 6.3.2 Lepelletier-Radsatz

Entsprechend der Verbrennungsmotorkomponente wird das Automatikgetriebe für die Berücksichtigung der 3D-Effekte bearbeitet. Dazu müssen die Momente der Drehträgheiten ( $J_1$ – $J_3$ ) und die Abstützmomente von den Bremsen gegen das Gehäuse berücksichtigt werden. In der Abb. 6.13 ist die Struktur des Lepelletier-Radsatz dargestellt.

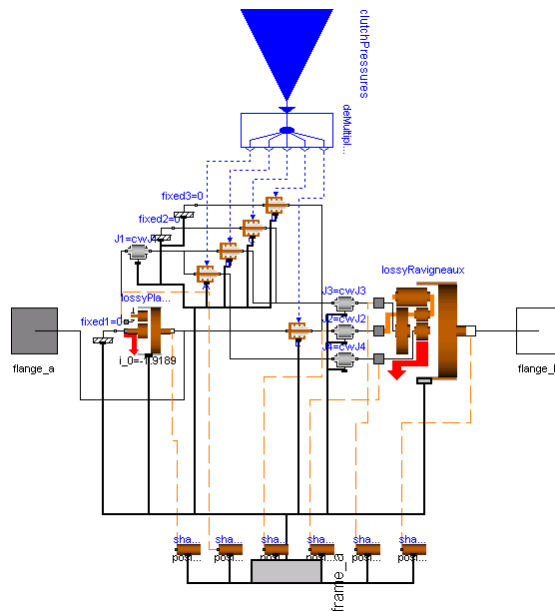


Abbildung 6.13: Objektdiagramm des Lepelletier-Radsatzes mit 3D-Effekten

## 7 Fahrdynamikmodell

Das Modell der Karosserie baut auf den komplexen Komponenten der Fahrdynamikbibliothek „VehicleDynamics“ Version 0.5 auf. Die „VehicleDynamics“-Bibliothek enthält für die Modellierung von unterschiedlichen Fahrdynamikmodellen notwendigen Komponenten. Aus Gründen des Umfangs kann in dieser Arbeit das Fahrdynamikmodell nicht ausführlich beschrieben werden. Deshalb werden ein Überblick über die Modellstruktur gegeben und die für die Verifizierung des Echtzeitmodells notwendigen Komponenten beschrieben. Eine ausführliche Beschreibungen der Fahrdynamik bieten die Fachbücher [Mit90, Mit95, Wil98, Ril94].

Am Ende dieses Kapitels wird ein neues Fahrdynamikmodell dargestellt, welches die 3D-Effekte des Antriebsstrangs berücksichtigt.

### 7.1 Koordinatensysteme

Zur anschaulichen Beschreibung der Kinematik des Fahrzeugs ist es zweckmäßig, als Grundlage ein System aus mehreren rechtsdrehenden Koordinatensystemen zu definieren. Damit können entsprechende Bewegungen bzw. Kräfte und Momente der einzelnen Komponenten im Raum beschrieben werden. In Abb. 7.1 ist eine mögliche Positionierung von Koordinatensystemen mit entsprechenden Bewegungsfreiheitsgraden dargestellt. Hierbei sind zur Übersichtlichkeit nicht alle Koordinatensysteme des *Modelica*-Fahrdynamikmodells dargestellt. Im Allgemeinen besitzen alle Komponenten ein separates Koordinatensystem. Durch die Verbindung der einzelnen Komponenten wird deren Lage im Raum festgelegt.

In der Abb. 7.1 werden folgende Koordinatensysteme festgelegt:

- ortsfestes Inertialsystem (Index  $I$ ),
- fahrbahnfestes Koordinatensystem (Index  $O$ ),
- fahrzeugfestes Koordinatensystem (Index  $F$ ) und
- radfestes Koordinatensystem (Indizes  $Rv1 \dots Rhr$ ).



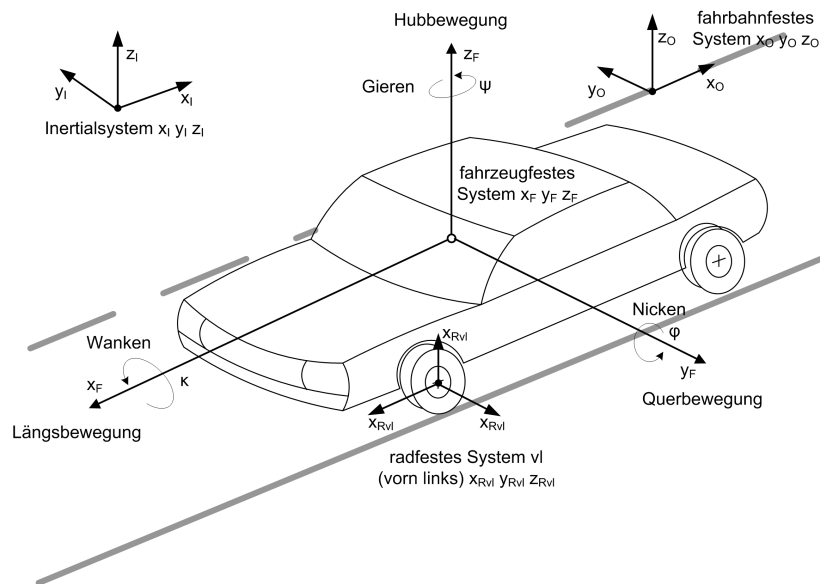


Abbildung 7.1: Bewegungsfreiheitsgrade eines Fahrzeugs und Lage der Koordinatensysteme im Raum [Hol01]

## 7.2 Struktur des Fahrdynamikmodells

Bei dem erstellten Modell handelt es sich um ein nichtlineares dynamisches Modell für zweiachsige Kraftfahrzeuge mit Differentialgetriebe, Antriebswellen und Bremsen. Das Fahrdynamikmodell ist hierarchisch aufgebaut und gliedert sich in die vier Ebenen:

1. *Vehicle level*
2. *Chassis level*
3. *Suspension level*
4. *Component level*.

Die oberste Ebene des Fahrdynamikmodells, das *Vehicle level*, ist in Abb. 7.2 dargestellt. In dieser Ebene wird die Fahrdynamikkomponente mit den anderen Fahrzeugkomponenten verbunden und die entsprechenden Komponenten für Antrieb, Karosserie und Umwelt definiert. Die nächste Hierarchiestufe bildet das *Chassis level*. Hier werden die typischen Fahrzeugkomponenten, wie Räder, Vorder- und Hinterradaufhängung und der Fahrzeuggrumpf festgelegt. Im *Suspension level* werden für die Vorder- und Hinterradaufhängungen die entsprechenden Feder-/Dämpferkomponenten und die Lenkung vereinbart. Die unterste Ebene bildet das *Component level*. Hier werden die einzelnen Komponenten für die Fahrdynamik modelliert. Dabei haben Komponenten, die die gleiche Aufgabe innehaben, gleiche Schnittstellen. Dadurch sind diese Komponenten sehr leicht austauschbar. In der *VehicleDynamics*-Bibliothek

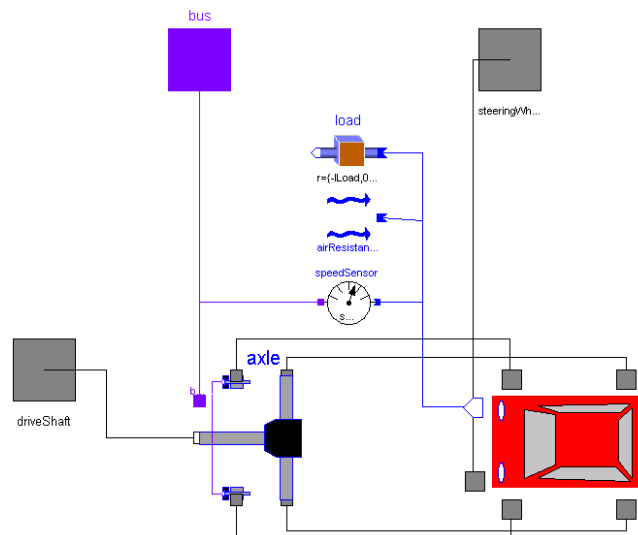


Abbildung 7.2: Objektdiagramm des Fahrdynamikmodells (*Vehicle level*)

sind verschiedene Komponenten, wie beispielsweise *Simple* und *MacPherson* für die einzelnen Radaufhängungen sowie verschiedene Lenk- und Stabilisatorsysteme, realisiert. Es können somit in einfachster Weise, ohne die erneute Modellierung einzelner Komponenten, verschiedene Fahrdynamikmodelle erstellt werden.

Durch den objektorientierten Modellierungsansatz kann das Fahrdynamikmodell nicht in die traditionellen aus der Regelungstechnik bekannten Teilmodelle

- Vertikaldynamik,
- Horizontaldynamik,
- Lenkung und
- Rad-Straße

aufgeteilt werden, sondern es wird ein MKS-Ansatz verwendet. Dabei wird jede Komponente für ihre spezifische Aufgabe erstellt. Erst durch die Verschaltung der einzelnen Komponenten zum Fahrdynamikmodell können alle für den Fahrbetrieb notwendigen Größen bestimmt werden. Selbstverständlich können die einzelnen Komponenten unabhängig verifiziert werden.

### 7.2.1 Chassis

Im Chassis sind alle Systeme wie Lenkung, Horizontaldynamik, Vertikaldynamik und Rad-Straße enthalten. Damit wird das komplette fahrdynamische Verhalten des Fahrzeugs abgebildet. Das in *Modelica* realisierte Chassismodell ist in der Abb. 7.3 dargestellt. Es besteht aus einer starren Karosserie, der den Fahrzeugschwerpunkt festlegt, einer Vorderradaufhängung

mit Lenkung und einer Hinterradaufhängung. Zudem sind die einzelnen Räder und Schnittstellen festgelegt. In der Schnittstelle *BGR* sind die räumlichen Kräfte und Momente, sowie die Position, die Geschwindigkeit und die Beschleunigung des Fahrzeugaufbaus definiert.

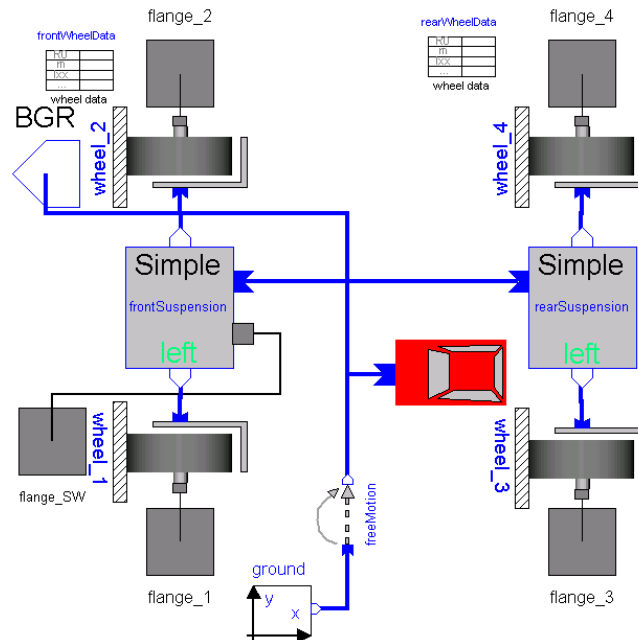


Abbildung 7.3: Objektdiagramm des Chassismodells (*Chassis level*)

Zum Modellverständnis und zur Beschreibung wichtiger fahrdynamischer Größen soll hier kurz die Horizontaldynamik beleuchtet werden. Eine Trennung in Horizontaldynamik und Vertikaldynamik des Karosseriemodells ist im realisierten *Modelica*-Modell jedoch nicht möglich.

Die Horizontaldynamik beschreibt die Bewegung des Fahrzeugs parallel zur Fahrbahnoberfläche. Zum Modellverständnis werden kurz an einem Zweispurmodell nach [Mit90] wichtige Modellgrößen erläutert. Der Aufbau des Fahrzeugs wird als starrer Körper mit der Masse  $m$  betrachtet. Der Schwerpunkt  $SP$  des Aufbaus bewegt sich mit einer tangential zur Bahnkurve gerichteten Geschwindigkeit  $v$  um einen Krümmungsmittelpunkt  $M$ , siehe Abb. 7.4. Der Winkel zwischen der Richtung der Fahrzeuggeschwindigkeit und der  $x_F$ -Achse des fahrzeugfesten Koordinatensystems wird im Allgemeinen als Schwimmwinkel  $\beta$  bezeichnet. Zusätzlich führt das Fahrzeug eine Drehbewegung mit der Gierwinkelgeschwindigkeit  $\dot{\psi}$  um seine durch den Schwerpunkt verlaufende Hochachse aus. Das Fahrzeug bewegt sich entsprechend dem Krümmungsradius  $\rho$  auf einer Bahnkurve. Am Fahrzeugaufbau greifen die Zentripetalkraft  $F_{Zpt,SP}$  und Luftkraft  $F_L$  an. Des Weiteren wirken über die Räder Radkräfte auf das Fahrzeug ein. Die Radkräfte werden in der Komponente Rad-Straße berechnet. Eine weitere wichtige Größe ist der Radeinschlagswinkel  $\delta$  an den einzelnen Rädern, der den Fahrerwunsch für

einen entsprechenden Fahrzeugkurs repräsentiert. Aus den Fahrzeugbewegungen  $v$  und  $\dot{\psi}$  sowie den Beschleunigungen  $\dot{v} = a$  und  $\ddot{\psi}$  lässt sich das Verhalten des Fahrzeugs in horizontaler Richtung bestimmen.

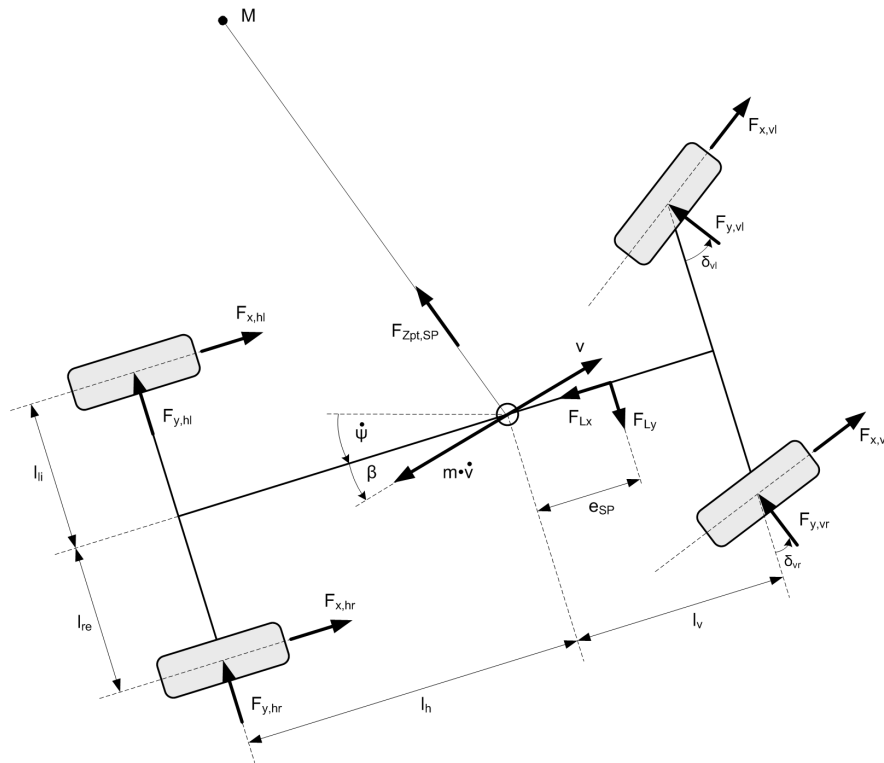


Abbildung 7.4: Geometrien und Kräfte am Zweispurmodell [Mit90]

## 7.2.2 Fahrwerk mit Lenkung

Das Fahrverhalten des Fahrzeugs wird im Wesentlichen durch das Fahrwerk bestimmt. Dazu sind in der „VehicleDynamics“-Bibliothek verschiedene Fahrwerke realisiert. Für die Echtzeitsimulation wird ein Fahrwerk, bestehend aus Vorderachse mit Lenkung und Hinterachse, verwendet, das durch die verwendeten Radaufhängungen „Simple“ die kürzeste Rechenzeit bietet. In der Abb. 7.5 ist das Vorderachsmodell, bestehend aus linker und rechter Radaufhängung, Lenkung und Stabilisator, dargestellt. Die Schnittstelle **frame\_C** stellt eine starre Verbindung von der Achse zur Karosserie her. Darüber hinaus sind zwei Schnittstellen **frame\_U\_1** und **frame\_U\_2** zum Anschluss der Räder und eine Schnittstelle für die Lenkung modelliert. Das Hinterachsmodell enthält außer der Lenkung die gleichen Komponenten. Eine Parametrisierung der Achskomponenten erfolgt durch den Datenbaustein **data**. Im Folgenden werden die Komponenten für Radaufhängung und Lenkung beschrieben.

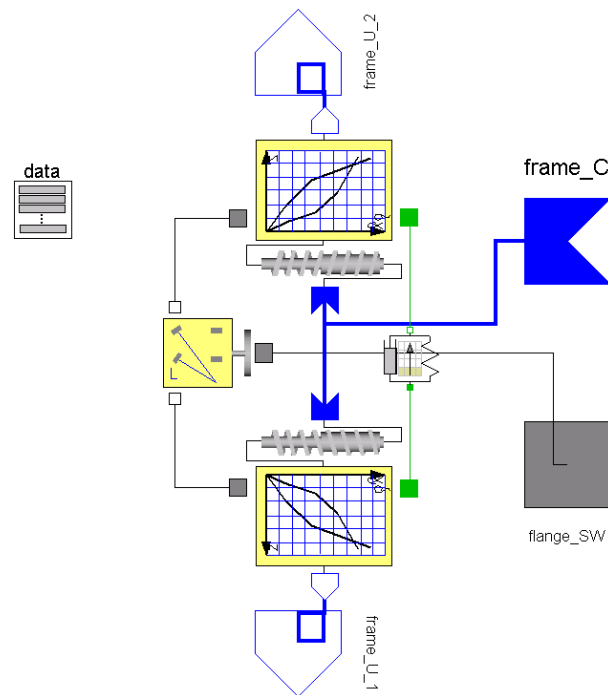


Abbildung 7.5: Objektdiagramm der vorderen Radaufhängungen mit Lenkung

### Radaufhängung

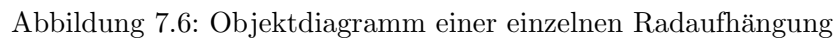
Das Modell der Radaufhängung bestimmt in erster Linie die Vertikaldynamik des Kraftfahrzeugs. Dadurch können die Aufbaubeschleunigungen und die dynamischen Radlastschwankungen beurteilt werden. In der Abb. 7.6 ist die *Modelica*-Komponente der Radaufhängung dargestellt.

Die wesentlichsten Komponenten der Radaufhängung sind:

- Ein masseloses Federdämpfermodell; Dabei werden die Feder und der Dämpfer jeweils durch eine nichtlineare Kennlinie charakterisiert.
- Ein Modell für den Lenkwinkel; Der Lenkwinkel am Rad wird durch den von außen aufgeprägten Lenkwinkel und einem Vorspurwinkel, der abhängig des Federwegs ist, berechnet.
- Ein Modell für den Radsturz; Dieser wird ebenfalls in Abhängigkeit des Federwegs berechnet.
- Eine Masse für die gesamte Radaufhängung.

### Lenkung

Die Lenkung eines Kraftfahrzeuges dient dazu, einen vom Fahrer vorgegebenen Lenkradwinkel  $\delta_L$  in die entsprechenden Einschlagwinkel der Räder  $\delta_{vl}$  und  $\delta_{vr}$  umzusetzen. Die



### 7.2.3 Reifen

Wenn Echtzeitfähigkeit gefordert oder eine detaillierte Nachbildung des physikalischen Reifenmodells nicht notwendig ist, werden so genannte makroskopische Reifenmodelle eingesetzt. Bei ihnen wird der Kontakt zwischen Reifen und Fahrbahn mittels mathematischer Modelle nachgebildet. Zur Begriffsklärung sind in der Abb. 7.8 die Radgeometrie und die am Reifen

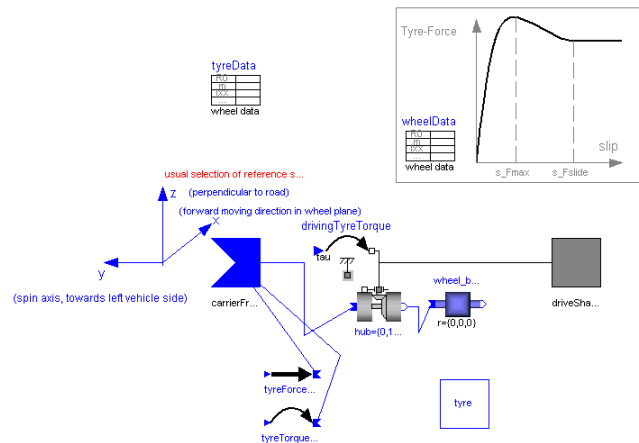


Abbildung 7.7: Objektdiagramm der Radkomponente mit dynamischem Reifenmodell

auftretenden Geschwindigkeiten, getrennt für den Antriebs- und den Bremsfall, dargestellt.

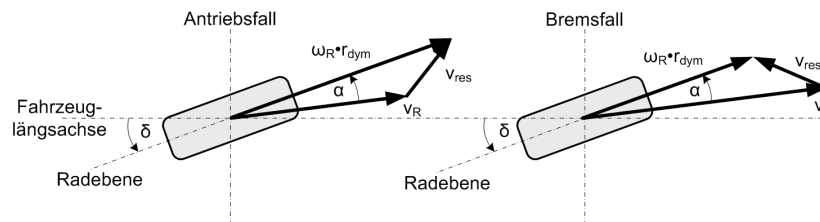


Abbildung 7.8: Geometrie und auftretende Geschwindigkeiten am Rad [Hol01]

Aufgrund der durch die Kraftübertragung zwischen Reifen und Fahrbahn auftretenden Gleitvorgänge und elastischen Verformungen des Reifengummis verläuft die Geschwindigkeit der Radaufstandsfläche  $v_R$  bei einer Kurvenfahrt im Gegensatz zur Radumfanggeschwindigkeit  $\omega_R r_{dym}$  nicht in Richtung der Radebene. Es bilden sich ein so genannter Schräglaufwinkel  $\alpha$  und die resultierende Geschwindigkeit  $v_{res}$  aus. Für die weitere Betrachtung ist das Verhältnis zwischen resultierender Geschwindigkeit und Absolutgeschwindigkeit, welche als Schlupf  $\lambda$  bezeichnet wird, von Interesse. Beim Schlupf wird zwischen Brems- und Antriebsschlupf unterschieden. Mit dem Schlupf wird dann meist aus einer gegebenen Reibwert-Schlupf-Kennlinie ein Reibwert  $\mu$  bestimmt. Mit dem Reibwert können dann die entsprechenden Radkräfte und -momente berechnet werden.

## 7.3 Abtrieb

Der Abtrieb beinhaltet die Welle vom Getriebe zum Hinterachsdifferential, das Hinterachsdifferential, die Wellen zu den Rädern sowie die Bremsen des Fahrzeugs. Auf der Eingangsseite

enthält der Abtrieb eine Schnittstelle zum Anschluss an das Getriebe und ausgangsseitig vier Schnittstellen für den Anschluss an die einzelnen Räder des Chassis. Für das Fahrdynamikmodell und die Echtzeitsimulation wurden Abtriebssysteme mit unterschiedlichen Brems- und Antriebskonfiguration realisiert. In der Abb. 7.9 ist das zur Verifizierung des Echtzeitmodells verwendete Abtriebsmodell mit zwei Bremsen an der Vorderachse sowie Heckantrieb dargestellt.

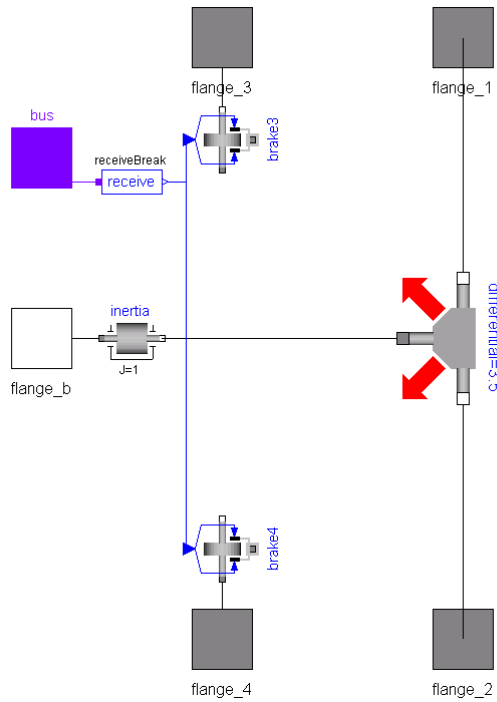


Abbildung 7.9: Objektdiagramm der Abtriebskomponente

Die Wellen werden idealisiert als starr angenommen und durch eine Drehträgheit repräsentiert, siehe Abschnitt 6.1.1. Das Differentialgetriebe teilt das Antriebsmoment vom Getriebe  $M_G$  auf das linke und rechte Antriebsradmoment auf,  $M_{Rl}$  beziehungsweise  $M_{Rr}$ . Das Differentialgetriebe enthält keine Trägheiten. Über eine geschwindigkeitsabhängige Verlustetabelle kann die Verlustleistung, die durch die Reibung in dem Zahnradgetriebe auftritt, definiert werden. Durch entsprechende Parametrierung werden im vorliegenden Modell wegen der Echtzeitfähigkeit jedoch keine Verluste betrachtet. Mit der Getriebeübersetzung  $i$  und den entsprechenden Winkeln  $\varphi$  vereinfachen sich die Systemgleichungen für das Differential zu:

$$\varphi_G = (\varphi_{Rl} + \varphi_{Rr}) \frac{i}{2} \quad (7.1)$$

$$M_{Rl} = M_{Rr} \quad (7.2)$$

$$-i \cdot M_G = M_{Rl} + M_{Rr} \quad (7.3)$$



Ausgehend von der Bremspedalstellung ist es die Aufgabe des Bremssystems, die an den Rädern des Fahrzeugs wirkenden Bremsmomente  $M_B$  zu erzeugen. Die Bremsen an den Rädern werden direkt über den Bus angesteuert. Dabei wird das Hydrauliksystem einfach durch das normierte Bremssignal im Bus ersetzt. Das vom Fahrer vorgegebene normierte Bremssignal wird somit ohne Verzögerung in eine Bremskraft am Rad umgesetzt.

Das Grundprinzip des Modells der Bremsen beruht auf dem gleichen hybriden Ansatz wie im Kupplungsmodell, siehe Abschnitt 6.2.3. Die Parameter des Modells und die Eingangsgröße unterscheiden sich jedoch. Die Geometriekonstante  $c_{geo}$  und der Reibwertkoeffizient  $\mu$  werden als feste Parameter angegeben. Die Normalkraft  $F_N$  berechnet sich aus der maximalen Bremskraft  $F_{Nmax}$  und dem normierten Bremssignal  $b$  zu:

$$F_n = F_{Nmax} \cdot b \quad . \quad (7.4)$$

Das Bremsmoment  $M_B$  ergibt sich demzufolge zu:

$$M_B = \begin{cases} 0 & : \text{ free} \\ M_{flange.b} = -M_{flange.a} & : \text{ locked} \\ c_{geo} F_n \mu & : \text{ forward} \\ c_{geo} F_n (-\mu) & : \text{ backward} \end{cases} \quad . \quad (7.5)$$

## 7.4 Umwelt

Das Umweltmodell besteht aus den beiden Komponenten Luft und Straße. Für das Echtzeitmodell wird die Komponente Straße durch eine ebene Fläche ohne Steigung und mit einem Reibwert  $\mu$  für die entsprechenden Fahrbahnabschnitte ausgeführt. In der Komponente Luft werden die Windgeschwindigkeit und die Luftdichte festgelegt, die für die Berechnung des Luftwiderstandes notwendig sind. Die Berechnung des Luftwiderstandes erfolgt in der Komponente Chassis und ist abhängig vom Quadrat der effektiven Anströmgeschwindigkeit, der Luftdichte, der Fläche und der Form des Körpers.

## 7.5 Geschwindigkeitsmessung

Die Geschwindigkeit des Fahrzeugs wird aus der Bewegung in x-Richtung des Fahrzeugmittelpunkts (*BGR*-Flansch) bestimmt. Das Koordinatensystem ist fahrzeugfest wobei die x-Achse in Fahrzeuginnenrichtung zeigt. Der Schwerpunkt des Fahrzeugs liegt im vorderen Drittel, wodurch die x-Achse leicht in die Fahrbahn zeigt. Zudem wird das Koordinatensystem durch Nick-, Wank- und Gierbewegungen des Fahrzeugs verdreht. Für die Geschwindigkeitsbestimmung entsteht somit ein Fehler, der aufgrund seiner geringen Signifikanz vernachlässigt wer-

den kann. Zu beachten ist deshalb, dass die absolute Geschwindigkeit des Fahrzeugs von der Geschwindigkeit „vehicleSpeed“ im *bus* abweicht.

## 7.6 Fahrdynamikmodell mit 3D-Effekten

Für die Untersuchung der 3D-Effekte im Antriebsstrang wurde auf der Grundlage des oben beschriebenen Modells ein neues Fahrdynamikmodell modelliert. Dafür muss die neue „VehicleDynamics“-Bibliothek Version 0.8 verwendet werden, die auf der „MultiBody“-Bibliothek Version 1.0 aufbaut. In der Abb. 7.10 ist die neue Struktur des Fahrdynamikmodells mit den neu modellierten Komponenten Karosserie, Abtrieb und Rad dargestellt.

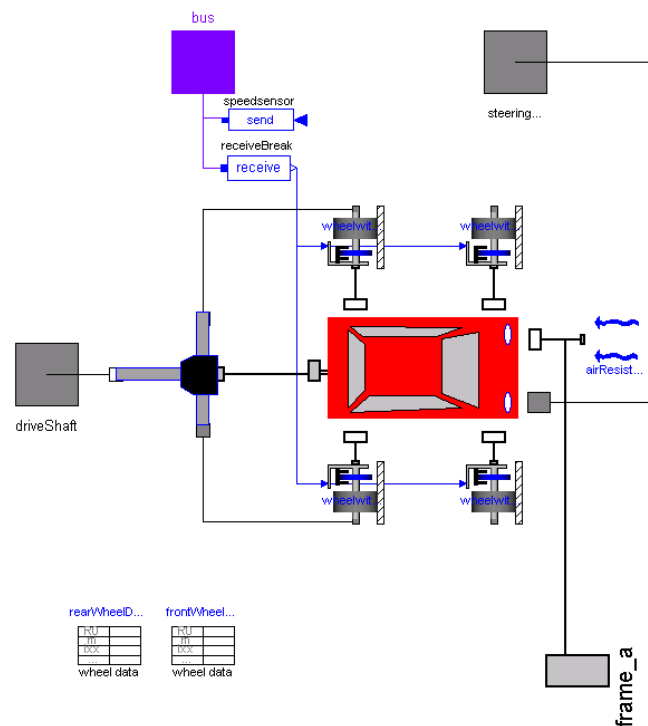


Abbildung 7.10: Objektdiagramm des Karosseriemodells mit 3D-Effekten

In der Komponente Abtrieb ist die Antriebswelle und das Differentialgetriebe vereinbart. Die Drehmomente im Antriebsstrang werden über die 1D-Schnittstellen ausgetauscht. Des Weiteren werden die Kreismomente der Antriebswelle berücksichtigt. Das Karosseriemodell wurde verändert, damit ein leichter Austausch der Räder mit Bremsen erfolgen kann. In den Radkomponenten werden die 3D-Abstützmomente der Bremsen direkt an der Radaufhängung befestigt. Weiter werden über die 3D-Schnittstelle (BGR) die Momente der Antriebsstrangkomponente an dem fahrzeugfesten Koordinatensystem befestigt.

## 8 Fahrermodell

Wie im Abschnitt 2.3 beschrieben, ist für die Echtzeitsimulation ein gesamtheitliches System, basierend aus Fahrzeug mit Umwelt und Fahrer, zu modellieren. Das Modell des Fahrers besitzt die Aufgabe, den menschlichen Fahrer nachzubilden. In einem klassischen Kraftfahrzeug mit Automatikgetriebe beeinflusst der Fahrer die Beschleunigung bzw. Verzögerung in Fahrzeuglängsrichtung über die vier Stelleinrichtungen:

- Fahrpedal,
- Bremspedal,
- Gangwahlhebel und
- eventuell ein Schaltelement zur direkten Gangwahl.

Außerdem nimmt der Fahrer über das Lenkrad Einfluss auf die Horizontalbewegung des Kraftfahrzeugs. Er beeinflusst über die entsprechenden Stelleinrichtungen als „intelligentes Regelungssystem“ das Fahrverhalten, d. h. er fährt mit einer gewählten Fahrzeuggeschwindigkeit einen Kurs ab.

Für das Gesamtfahrzeugmodell wurde ein Fahrermodell, bestehend aus einer Regelung für die Fahrzeuggeschwindigkeit sowie einer Steuerung für den Fahrzeugkurs, erstellt. Im Folgenden werden die einzelnen Fahrerkomponenten beschrieben.

### 8.1 Longitudinal-Fahrer

Die Aufgabe der Geschwindigkeitsregelung in Fahrzeuglängsrichtung übernimmt der Longitudinal-Fahrer. Über diesen wird die Regelstrecke der Fahrzeuglängsgeschwindigkeit geschlossen. Für das Echtzeitmodell wurde bestehend auf dem Fahrermodell der „PowerTrain“-Bibliothek ein vereinfachter Fahrerregler entwickelt, siehe Abb. 8.1.

Im Kern besteht dieser aus einem PI-Regler, der die Fahr- und Bremspedalstellung für die gewünschte Geschwindigkeit berechnet. Mittels Kennfeld werden für das Fahrmanöver die Sollgeschwindigkeit, die Stellung des Gangwahlhebels und der geforderte Gang vorgegeben. Die Modellausgänge Drosselklappenstellung, Bremse sowie Anlasser, Gangwahlhebel

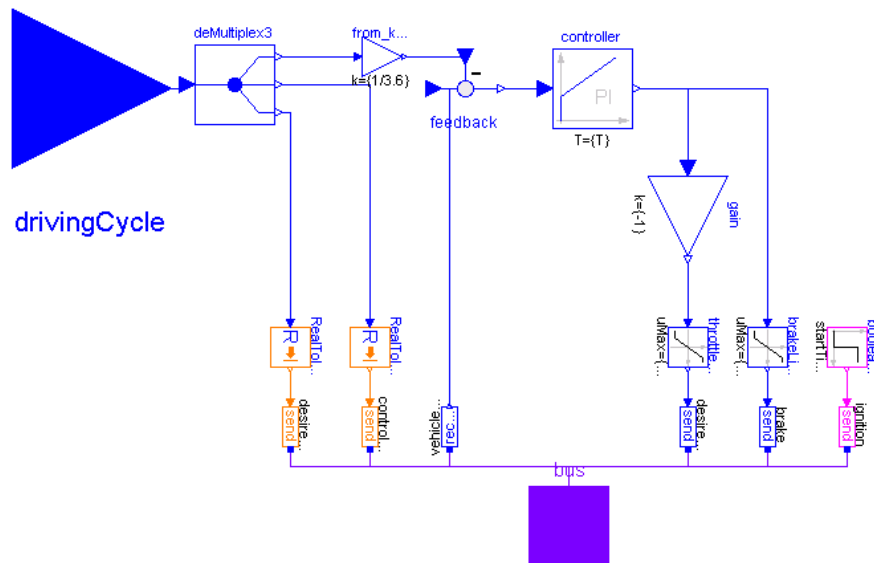


Abbildung 8.1: Objektdiagramm des Longitudinal-Fahrers

und vorgegebener Gang werden als normierte Signale direkt auf den Bus gegeben. Durch die Normierung des Brems- und Drosselklappensignals wird eine gleichzeitige Betätigung der Stelleinrichtungen ausgeschlossen.

Das Fahrermodell stand in dieser Arbeit nicht im Vordergrund. Der Longitudinal-Fahrer soll dennoch bei geringen Änderungsgeschwindigkeiten der Regelgrößen eine geringe Abweichung zwischen Fahrzeug-Ist- und Fahrzeug-Sollgeschwindigkeit gewährleisten. Dazu sind die Verstärkung  $k$  und die Zeitkonstante  $t$  des PI-Reglers entsprechend zu parametrisieren. Außerdem ist auf einen angemessenen Fahrzyklus zu achten, d. h. der vorgegebene Fahrzyklus darf keine großen Sprünge enthalten. Im Kapitel 9 werden für die Echtzeitsimulation eigene Fahrzyklen definiert. Bei ungünstiger Parametrisierung des Longitudinal-Fahrers kann es sonst zu Pendelschaltungen kommen.

## 8.2 Lateral-Fahrer

Die Steuerung des Fahrzeugkurses übernimmt der Lateral-Fahrer. Aufbauend auf dem Fahrermodell der „VehicleDynamics“-Bibliothek wurde eine vereinfachte Lenksteuerung modelliert, siehe Abb. 8.2. Mittels Kennlinie oder einer parametrisierbaren Sinusschwingung kann ein Lenkwinkelverlauf vorgegeben werden.

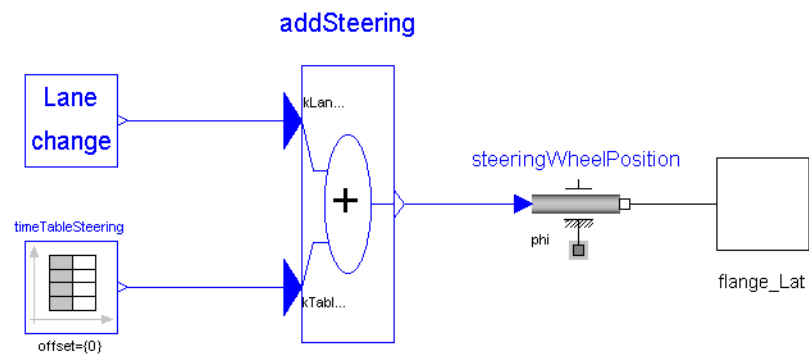


Abbildung 8.2: Objektdiagramm des Lateral-Fahrers

## 9 Detailliertes Fahrzeug-Echtzeitmodell

Aus den in den Kapiteln 6, 7 und 8 beschriebenen Fahrzeugkomponenten wird in diesem Kapitel das echtzeitfähige Gesamtfahrzeugmodell erstellt. Insbesondere werden die für die Echtzeitsimulation wichtige Parametrisierung und Initialisierung des Modells beschrieben.

### 9.1 Gesamtfahrzeugmodell

Für die Echtzeitsimulation des Gesamtfahrzeugmodells wurde, aufbauend auf den Bibliotheken „PowerTrain“ und „VehicleDynamics“, eine separate Bibliothek „PowerVehicle“ erstellt. Die Bibliothek enthält alle zum Testen und zur Echtzeitsimulation des Gesamtmodells notwendigen veränderten und neuen Komponenten. Der Schwerpunkt lag dabei nicht auf einer neuen Modellierung des Fahrzeugs, sondern auf der Integration der Komponenten Fahrermodell, Antriebsmodell und Fahrdynamikmodell zu einem echtzeitfähigen Gesamtfahrzeugmodell. In der Abb. 9.1 ist die oberste Ebene des detaillierten Fahrzeugmodells dargestellt.

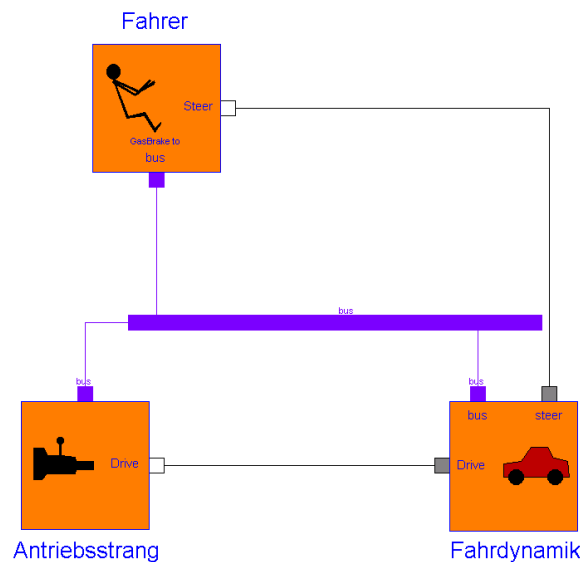


Abbildung 9.1: Objektdiagramm des detaillierten Fahrzeugmodells

In dieser Ebene wird aus den einzelnen Teilsystemen Antriebsstrang, Fahrdynamik und Fahrer das Gesamtsystem gebildet. Der Antriebsstrang besteht aus einem Verbrennungsmotor

mit Anlasser und einem Sechsgang-Automatikgetriebe, siehe Kapitel 6. Aus Gründen der Echtzeitfähigkeit sind nur zwei Bremsen an der Vorderachse modelliert. In das Fahrzeug ist der Antriebsstrang längs eingebaut und treibt die Hinterachse an. Das Fahrdynamikmodell besteht aus einer starren Karosserie und zwei Achsen mit Rädern, wobei die Vorderachse lenkbar ist, siehe Kapitel 7. Das Fahrermodell besteht aus den im Kapitel 8 beschriebenen Komponenten Longitudinal- und Lateral-Fahrer.

## 9.2 Parametrisierung

Nach dem Erstellen des Gesamtfahrzeugmodells erfolgt die Parametrisierung des Modells. Mittels einer Kombination aus theoretischer und experimenteller Vorgehensweise werden sämtliche benötigten Modellparameter ermittelt. Alle Parameter des Modells sind zeitinvariant. Im Folgenden wird ein Parametersatz für ein fiktives Fahrzeug betrachtet. Dabei ist besonders auf eine konsistente und vorteilhafte Parametrisierung des Modells zu achten. Für die Echtzeitsimulation ist auf Grund der Schrittweite von  $t = 1$  ms das Gesamtfahrzeugmodell so zu initialisieren, dass bei Beginn der Echtzeitsimulation die Rechenzeit eingehalten wird. Die benötigten Parameter lassen sich, je nach Art ihrer Ermittlung, in drei Klassen einteilen [Hol01]:

1. Konstruktiv bekannte oder statisch messbare Parameter
2. Aus Messverfahren schätzbare Parameter
3. Durch Modellabgleich bestimmbare Parameter.

Konstruktiv bekannte oder statisch messbare Parameter können im Allgemeinen aus den vom Fahrzeughersteller zur Verfügung gestellten Datenblättern ermittelt werden. Zu diesen Parametern gehören geometrische Abmessungen des Fahrzeugs (z. B. Fahrzeuglänge und Radstand), makroskopisch beschriebene Baugruppen (z. B. Getriebeübersetzungsverhältnis, Feder- und Dämpferkennlinien) sowie die in Steuergeräten implementierte Software (z. B. Schaltkennlinien für Automatikgetriebe). Zusätzlich sind in diese Gruppe Parameter begriffen, die mit einem vertretbaren Aufwand messtechnisch ermittelt werden können (z. B. Fahrzeugmasse, Luftwiderstandsbeiwert und Motorkennfeld).

Darüber hinaus ist es möglich, einige Parameter mit Hilfe von Schätzverfahren zu bestimmen. Diese Verfahren sind im Allgemeinen jedoch gegenüber direkten Messverfahren mit größeren Fehlern behaftet. Deshalb sollten diese nur eingesetzt werden, wenn Parameter nicht konstruktiv bekannt oder direkt gemessen werden können. Diese Verfahren werden oft auch bei zeitlich veränderlichen Parametern angewendet.

Nicht alle Modellparameter sind konstruktiv bekannt, messbar oder schätzbar. Deshalb bleiben modellbedingt Parameter übrig, die nicht ermittelt werden können. Diese Parameter werden dann durch einen Modellabgleich bestimmt, d. h. durch Abgleich von Messfahrten und entsprechenden Simulationsergebnissen werden diese Parameter empirisch eingestellt. Dies führt oft zu Mehrgrößenoptimierungsproblemen [Hol01], die nur durch entsprechendes Expertenwissen und mit nichtlinearen Optimierungsverfahren gelöst werden können.

Für das detaillierte Gesamtfahrzeugmodell müssen mehr als 10000 Parameter bestimmt werden. Deshalb wird aus Gründen des Aufwands und der Datenbeschaffung im wesentlichen die Grundparametrisierung der Komponenten aus den Bibliotheken verwendet. Bei der Simulation hat sich gezeigt, dass jedoch einige Daten leicht verfälscht sind, wodurch einige Simulationsergebnisse von der Realität abweichen können. Im Folgenden werden daher die Parameter beschrieben, die für die Echtzeitsimulation angepasst wurden.

Das Drehmoment des Verbrennungsmotors wird mit dem in der Abb. 9.2 dargestellten Kennfeld beschrieben. Das maximale Drehmoment von  $M_{\text{mot,max}} = 320 \text{ Nm}$  steht bei der Motordrehzahl  $n_{\text{mot}} = 4000 \text{ min}^{-1}$  zur Verfügung. Die Motorsteuerung regelt die Leerlaufdrehzahl auf  $n_{\text{mot,leer}} = 1200 \text{ min}^{-1}$  und begrenzt die Motordrehzahl auf  $n_{\text{mot,max}} = 5600 \text{ min}^{-1}$ . Der Motor wird mit dem Anlasser über eine Zeit von  $t_{\text{anl}} = 0.5 \text{ s}$  und mit einem Moment von  $M_{\text{anl}} = 150 \text{ Nm}$  gestartet.

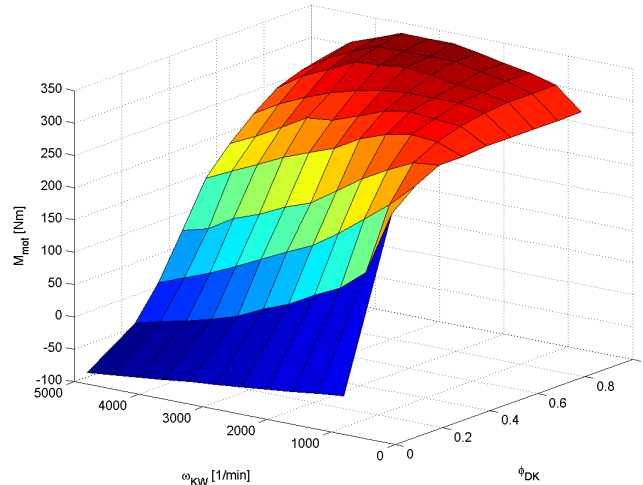


Abbildung 9.2: Motormoment-Kennfeld des Verbrennungsmotors

Die Übersetzungsverhältnisse der einzelnen Gänge des Sechsgang-Automatikgetriebes wurden aus der „PowerTrain“-Bibliothek übernommen. Aus Gründen der Echtzeitfähigkeit besitzt das Automatikgetriebe keine Verluste. Die Trägheitsmomente der Wellen im Antriebstrang wurden für die Echtzeitsimulation angepasst, siehe Tabelle 9.1. Bedingt durch die realisierte Steuerung der Kupplungen und Bremsen im Getriebe entstehen, infolge hoher Dreh-



zahldifferenzen in den Schaltelementen, große Momente im Antriebsstrang. Dies führte bei der Simulation zu mehreren Zustandsumschaltungen in den Schaltelementen, wodurch die Echtzeitbedingung von  $t < 1$  ms nicht eingehalten wurde. Zur Lösung dieses Problems wurde der Parameters  $\omega_{\text{small}} = 50$  rad/s, der das Umschalten von einem Gleitzustand direkt in den entgegen gesetzten Gleitzustand bewirkt, angepasst.

Trägheitsmoment	Formelzeichen	in kg m <sup>2</sup>
Motor	$J_M$	0.15
Pumpenrad des Drehmomentwandlers	$J_{WP}$	0.75
Turbinenrad des Drehmomentwandlers	$J_{WT}$	0.75
Getriebeachsen 1 bis 4	$J_{G1 \dots G4}$	0.01
Antriebsachse	$J_A$	0.2
Räder 1 bis 4	$J_{R1 \dots R4}$	1

Tabelle 9.1: Trägheitsmomente des Antriebstrangs

In der Getriebesteuerung werden die Schaltkennlinien für die Hoch- und Rückschaltung sowie die Druckverläufe für die Schaltelemente festgelegt. Für die Antriebsstrang-Konfiguration, bestehend aus Motor, Automatikgetriebe, Differential und Rädern wurde das Schaltkennfeld, siehe Abb. 9.3, empirisch bestimmt. Jede Kennlinie wird durch vier Stützpunkte charakterisiert. Dadurch sind einerseits ein wirtschaftlicher Betrieb und andererseits eine maximale Beschleunigung gewährleistet. Außerdem werden durch die Spreizungen der Schaltkennlinien Pendelschaltungen vermieden.

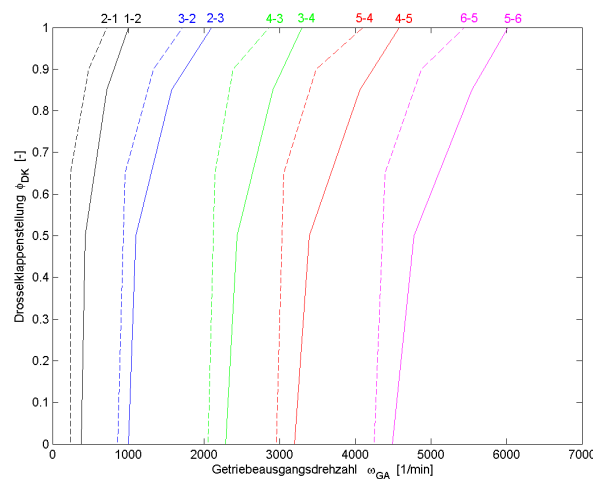


Abbildung 9.3: Hoch-(volle Linien) und Rückschaltkennlinien (gestrichelte Linien) des Automatikprogramms

Basierend auf den optimierten Druckverläufen für einen Antriebsstrang mit demselben Sechsgang-Automatikgetriebe werden die Druckverläufe der Schaltelemente für das Fahrzeugmodell empirisch eingestellt, siehe Abschnitt 6.2.2. In der Tabelle 9.2 sind die Parameter der Schaltelemente (SE), die den Druckverlauf bestimmen, dargestellt.

Parameter	SE A	SE B	SE C	SE D	SE E
<b>maxPressure</b>	40.047 bar	31.924 bar	42.84 bar	33.473 bar	69.217 bar
<b>riseTime</b>	0.68716 s	0.69779 s	0.75314 s	0.48023 s	0.5617 s
<b>fallTime</b>	0.38038 s	0.39204 s	0.42676 s	0.40209 s	0.75982 s
<b>T</b>	0.001 s	0.001 s	0.001 s	0.001 s	0.001 s

Tabelle 9.2: Trägheitsmomente des Antriebstrangs

Das Einstellen der Schaltkennlinien und der Druckverläufe ist sehr aufwendig. Deshalb sind Expertenwissen und Optimierungsverfahren für die Parametrisierung von elektronischen Getriebesteuerungen notwendig. Durch die unterschiedlichen Antriebsstrang-Konfigurationen steigt zudem der Aufwand bei der Fahrzeugentwicklung stark an. In [Cim03, SOC04] wird das am DLR entwickelte Mehrkriterienoptimierungsverfahren *MOPS* (Multi-Objective Parameter Synthesis) für die Optimierung der Schaltkennlinien und Druckverläufe in einer HiL-Simulation beschrieben.

Die Fahrzeugmasse mit Fahrer und Zuladung beträgt 1800 kg und liegt im Mittelpunkt des fahrzeugfesten Koordinatensystems. Die Bremsen des Fahrzeugs werden jeweils mit einer maximalen Kraft von  $F_B = 2500$  N betrieben, dies entspricht einem maximalem Bremsmoment an einer Bremse von  $M_B = 1250$  Nm. Das Differentialgetriebe wurde mit einem Übersetzungsverhältniss  $i = 3.5$  definiert. Die Parameter für z. B. Feder-/Dämpferelemente und Rad (mit Reifen) des Fahrdynamikmodells wurden aus der „VehicleDynamics“-Bibliothek übernommen. Es wurde auf eine bessere Parametrisierung des Fahrdynamikmodells verzichtet, weil dies sehr aufwändig ist und für die Ergebnisse dieser Arbeit eine untergeordnete Rolle spielt.

Das Fahrzeugmodell ist so aufgebaut, dass eine Animation des Fahrzeugs und des Getriebes unterstützt wird. Die Berechnung der Animation ist jedoch rechenintensiv, weshalb für die Echtzeitsimulation auf eine Animation des Fahrzeugs verzichtet wird.

## 9.3 Initialisierung

Für die Echtzeitsimulation müssen die Variablen des Fahrzeugmodells entsprechend initialisiert werden, damit die Rechendauer für die ersten Simulationsschritte nicht zu groß wird.

Aufgrund der Komplexität des Fahrzeugmodells ist bei der Initialisierung besonders auf konsistente Daten zu achten. Eine ungünstige Initialisierung kann sonst wegen Ereignisiterationen zum Abbruch der Simulation führen.

Jede beim Simulationsstart vorgegebene Fahrzeuggeschwindigkeit bedingt eine neue Initialisierung der Winkelgeschwindigkeiten der einzelnen Antriebsstrangkomponenten. Außerdem müssen für alle unterschiedlichen Beladungszustände die Initialisierungswerte für die Höhe des Fahrzeugmittelpunkts sowie die Wege und Geschwindigkeiten der Feder-/Dämpferelemente neu bestimmt werden. Die Initialisierungswerte wurden für das oben parametrisierte Fahrzeugmodell und einer konstanten Fahrzeuganfangsgeschwindigkeit von  $v_{Fx} = 1.2 \text{ km/h}$  durch Simulation ermittelt. Bei der Simulation des detaillierten Fahrzeugmodells hat sich gezeigt, dass beim Starten aus dem Fahrzeugstillstand oder dem Abbremsen bis zum Fahrzeugstillstand ständige Zustandsänderungen in den Radkomponenten auftreten, wodurch ein großer numerischer Fehler auftritt. Dies konnte in der Offline-Simulation durch eine kleinere Schrittweite  $t < 0.5 \text{ ms}$  behoben werden. Mit dieser Schrittweite ist das Modell jedoch nicht mehr echtzeitfähig. Aus diesem Grund wird durch entsprechende Fahrzyklen der Fahrzeugstillstand vermieden.

In *Dymola* wird die Initialisierung des Modells gesondert behandelt. So wird für die Bestimmung der Initialwerte ein eigener Code generiert. Durch die symbolische Vorverarbeitung des Gleichungssystems entsteht eine effiziente und robuste Lösung der Initialisierungsgleichungen mit einer guten Diagnostik in Problemfällen. In *Dymola* bestehen zwei Möglichkeiten Variablen und Parameter zu initialisieren:

1. durch die zusätzlichen **Attribute** (Modifier) **start** und **fixed**
2. durch **Gleichungen** in einer „initial equation“-Sektion.

Mit dem Attribut **start** kann für Parameter der Startwert festgelegt werden. Das boolesche Attribut **fixed**, bestimmt ob der Startwert während der Initialisierung von *Dymola* geändert werden darf. In den Initialgleichungen können alle Variablen und Parameter des Modells beeinflusst werden. Die Initialisierungsgleichungen können dabei in allen Ebenen des Modells erfolgen. Aus Gründen der Übersichtlichkeit ist eine Initialisierung auf der obersten Ebene sinnvoll.

Der Antriebsstrang des Fahrzeugmodells wurde für die Echtzeitsimulation entsprechend der Tabelle 9.3 initialisiert. *Dymola* erstellt dann selbstständig mittels des Gleichungssystems alle notwendigen Initialwerte des Antriebsstrangs.

Initialwert	start		fixed
$\omega_{\text{Mot}}$	0	rad/s	true
$\omega_{\text{Wand,dif}}$	0	rad/s	false
$\omega_{\text{Get,A,mod}}$	0		true
$\omega_{\text{Get,A,dif}}$	-2.2	rad/s	true
$\omega_{\text{Get,B,mod}}$	0		true
$\omega_{\text{Get,B,dif}}$	0.75	rad/s	true
$\omega_{\text{Get,C,mod}}$	0		true
$\omega_{\text{Get,D,mod}}$	0		true
$\omega_{\text{Get,E,mod}}$	0		true
$\omega_{\text{Rad,1}}$	1.465	rad/s	true
$\omega_{\text{Rad,2}}$	1.465	rad/s	true
$\omega_{\text{Rad,3}}$	1.465	rad/s	true
$\omega_{\text{Rad,4}}$	1.465	rad/s	true

Tabelle 9.3: Trägheitsmomente des Antriebstrangs

# 10 Simulationsergebnisse

In diesem Kapitel werden die Simulationsergebnisse des detaillierten Fahrzeugmodells dargestellt. Bevor das Fahrzeug in einer Echtzeitsimulation untersucht wird, erfolgt mit dem Simulationssystem *Dymola* eine „Offline“-Simulation zur Verifizierung des Modells. Dabei wird für die jeweiligen Simulationssysteme die Durchführung erläutert.

Im Abschnitt 10.3 wird das detaillierte Fahrzeugmodell mit 3D-Effekten im Antriebsstrang dargestellt. Dafür werden die in den Abschnitten 6.3 und 7.6 dargestellten Komponenten verwendet. Weiter werden dieses Fahrzeugmodell mit dem Fahrzeugmodell aus dem Abschnitt 9.1 verglichen und ausgewählte Simulationsergebnisse dargestellt.

## 10.1 Verifizierung des Fahrzeugmodells

Nach der Modellerstellung und Parametrisierung erfolgte eine Verifizierung des detaillierten Fahrzeugmodells. Dazu wurden mit dem Simulationssystem *Dymola* verschiedene Fahrmanöver in einer SiL-Simulation durchgeführt. Die Simulationsergebnisse wurden dann mit Ergebnissen aus der Literatur verglichen. Über die Güte des Fahrzeugmodells kann keine genaue Aussage getroffen werden, weil bei der Parametrisierung die Echtzeitfähigkeit im Vordergrund stand. Somit können die Ergebnisse der simulierten Fahrmanöver leicht von der Realität abweichen.

### 10.1.1 Simulationsdurchführung mit Dymola

In diesem Abschnitt werden entsprechende Einstellungen für die Simulation mit *Dymola* beschrieben, die zur erfolgreichen Verifizierung sinnvoll sind. Für die Simulation eines *Modelica*-Modells erstellt *Dymola* entsprechend den Simulationseinstellungen aus der *Modelica*-Syntax einen simulierbaren C-Code. Mit diesem Code können anschließend in *Dymola* verschiedene Simulationen durchgeführt werden. Dabei können Parameteränderungen ohne Neuerstellung des Codes vorgenommen werden. Die Ergebnisse der Simulationen werden im .mat-Format mit dem Simulationsnamen abgespeichert und stehen zur nachträglichen Bearbeitung und Darstellung in *Dymola* und *Matlab* zur Verfügung.

In dem Fenster „Experiment Setup“ können leicht über die Register

- *General*,
- *Translation*,
- *Output*,
- *Debug*,
- *Compiler* und
- *Realtime*

die Simulationseinstellungen bearbeitet werden. Im Register *General* werden das verwendete Integrationsverfahren mit maximalem Fehler oder Schrittweite, das Ausgabeintervall und der Name der Simulation vereinbart. Im Register *Translation* werden Einstellungen für die Erstellung des Modell-Codes vorgenommen. Für die Erhöhung der Rechenperformance kann der Schalter *Evaluate* gesetzt werden. Damit werden die veränderbaren Parameter auf die oberste Ebene begrenzt. Dies bringt auch bei der Echtzeitsimulation Vorteile, weil nur die Parameter der obersten Ebene zwischen Haupt- und Zielrechner übertragen werden. Mit gesetztem *LogDefaultInitialConditions*-Schalter werden im „Log-Fenster“ alle vom *Dymola*-Interpreter initialisierten Variablen angezeigt. Durch einen Vergleich der initialisierten Variablen mit den Ergebnissen aus der Simulation lassen sich die Initialwerte bestimmen. Im Register *Output* können die Variablen und Parameter eingestellt werden, die während der Simulation gespeichert werden sollen. Für die Verifizierung des Modells ist es sinnvoll, alle Parameter des Modells zu speichern. Bei einer umfangreichen Simulation kann die Ergebnisdatei auf eine Größe von mehr als 100 MB anwachsen. Die Größe der Ergebnisdatei hängt außer vom Modell von der Simulationsdauer, den gespeicherten Variablen und dem Ausgabeintervall ab. Im Register *Debug* ist es für die Verifizierung vorteilhaft, die Schalter *LogEventsduringsimulation* und *LogEventsduringinitialization* zu setzen. Damit können die Events (Zustandsänderungen) während der Initialisierung und der Simulation leicht ermittelt werden.

Darüber hinaus können durch Texteingabe im „Log“-Fenster Simulationseinstellungen bearbeitet werden. Mit dem versteckten Schalter *Hidden.Define.PrecisionTiming = true* können für die Beurteilung der Rechengeschwindigkeit die durchschnittliche Rechenzeit sowie die Events und Modellauswertungen während eines Rechenschritts dargestellt werden.

### 10.1.2 Integrationsverfahren und Eventbehandlung

In einer umfangreichen Untersuchung wurden die Integrationsverfahren und die Schrittweiten verifiziert. Bei den ersten Simulationen wurde das Verfahren „dassl“ mit variabler Schrittwei-

te eingesetzt. Dieses Verfahren berücksichtigt die Zustandsumschaltungen im Modell, d. h. die Events werden genau bestimmt. Damit ist der Rechenfehler bei Auftreten eines Events sehr gering. Danach wurden für die Echtzeitsimulation notwendige Integrationsverfahren mit fester Schrittweite untersucht. Hierbei stellte sich heraus, dass das Integrationsverfahren „Explizit Euler“ mit der Schrittweite  $t = 1$  ms die beste Rechenperformance bei einem kleinen numerischen Fehler bietet. Wenn in einem Rechenschritt ein Event auftritt, erfolgt eine Zustandsumschaltung des Modells, d. h. die Modellgleichungen werden umgestellt. Um konsistente Daten zu gewährleisten, werden die neuen Modellgleichungen ausgewertet und alle Variablen neu berechnet. Dafür sind meist mehrere Modellauswertungen für einen Rechenschritt notwendig. Damit steigt während einer Zustandsumschaltung die Rechenzeit für einen Rechenschritt stark an. Bei der Simulation wurde allgemein eine Zunahme um den Faktor 2.5 gegenüber der mittleren Rechenzeit beobachtet. Weiter haben Untersuchungen gezeigt, dass bei einer Schrittweite  $t < 0.5$  ms der Rechenfehler minimiert werden könnte. Jedoch steigt damit die Rechenzeit stark an, was eine Echtzeitsimulation ausschließt. Bei der Simulation wird im Folgenden das Integrationsverfahren „Explizit Euler“ mit der Schrittweite  $t = 1$  ms eingesetzt.

### 10.1.3 Ausgewählte Simulationsergebnisse

Zur Überprüfung und Beurteilung des Fahrverhaltens von Fahrzeugen existieren viele verschiedene Fahrmanöver. Hierbei wird zwischen längsdynamischen Manövern zur Beurteilung des Schaltverhaltens, der Agilität und des Verbrauchs und querdynamischen Manövern zur Beurteilung des Kurven- und Wechselkurvenverhaltens unterschieden. Aus Gründen des Platzes werden exemplarisch ein Längsfahrzyklus und eine stationäre Kurvenfahrt für die Darstellung ausgewählt. Die entsprechenden Parameter für die unterschiedlichen Fahrmanöver werden in der obersten Ebene des Modells definiert. Im Folgenden werden ausgewählte Fahrmanöver zur Beurteilung des detaillierten Fahrzeugmodells dargestellt.

#### Längsfahrzyklus

In dem Längsfahrzyklus, d. h. der Lenkwinkel ist konstant  $\delta_L = 0$ , werden insbesondere die Auswirkungen des Antriebsstrangs auf die Längsdynamik untersucht. Zur Beurteilung der Längsdynamik wird ein vordefiniertes Geschwindigkeitsprofil abgefahren. Aus Gründen der Echtzeitfähigkeit wird die Simulation im Leerlauf mit der Anfangsgeschwindigkeit  $v_{Fx} = 1.2$  km/h und abgeschaltetem Motor gestartet. Der Verbrennungsmotor wird mit dem Starter angelassen. Nach  $t = 5$  s hat sich das Fahrzeugverhalten eingeschwungen und das Fahrmanöver beginnt.

In der Abb. 10.1 sind die Simulationsergebnisse des Fahrzeugmodells ohne Motormoment-

begrenzung bei Schaltvorgängen dargestellt. Die Soll- und Istgeschwindigkeit des Fahrzeugs sind im obersten Diagramm dargestellt. Darunter ist der Verlauf des Geschwindigkeitsfehlers abgebildet. Bei den Signalverläufen ist nach den Gangwechseln ein größerer Geschwindigkeitsfehler zu beobachten. In den weiteren Diagrammen sind der eingelegte Gang, die Längsbeschleunigung und der Schlupf am rechten Hinterrad zu sehen. Des Weiteren treten neben den größeren Geschwindigkeitsfehlern auch große Beschleunigungen in Fahrzeuginnenrichtung und Radschlupfe auf. Dies ist auf das Motormoment während der Schaltung und auf die unterschiedlichen Motordrehzahlen vor und nach der Schaltung zurückzuführen.

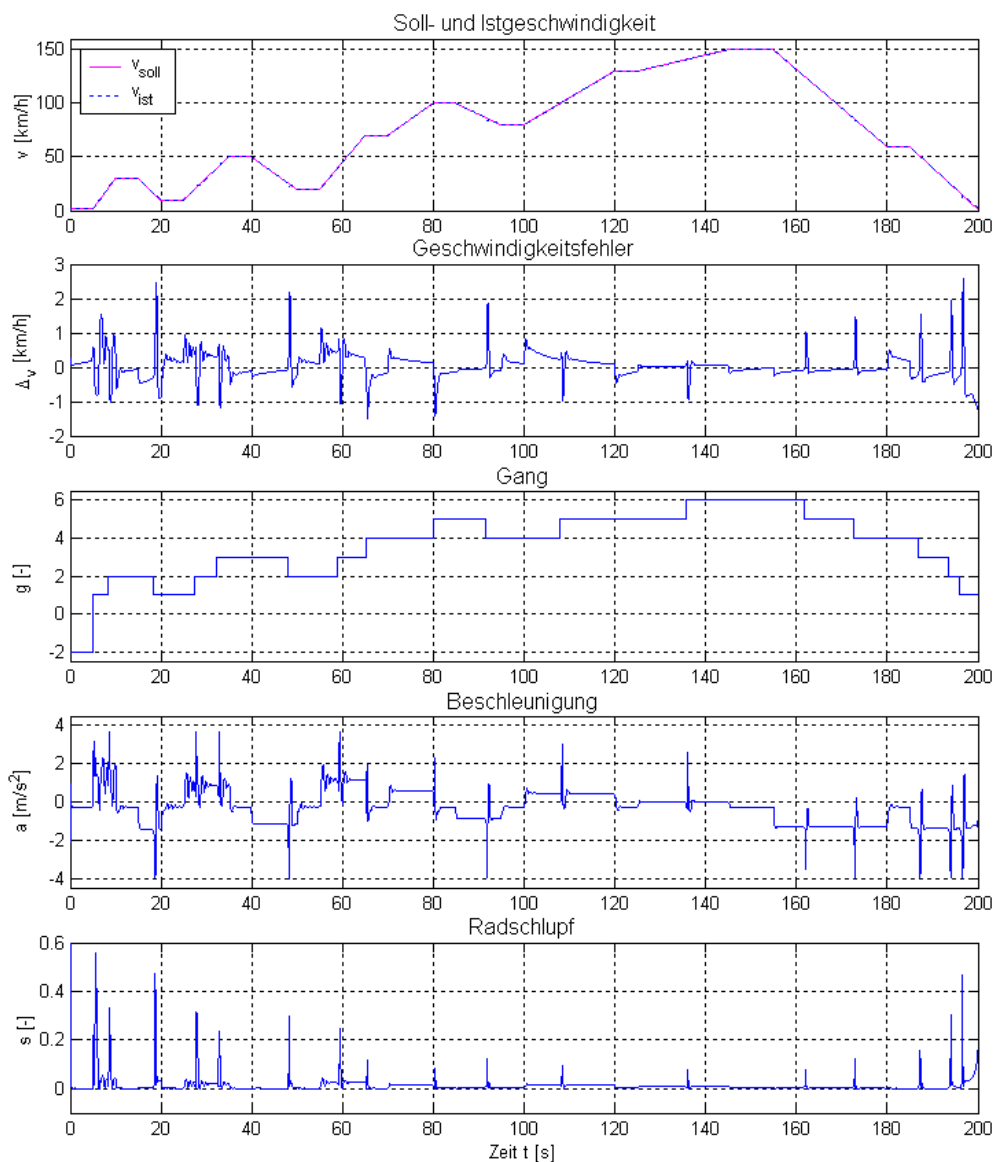


Abbildung 10.1: Ausgewählte Offline-Simulationsergebnisse des Längsfahrzyklus ohne Steuerung des Motormoments

Die kurzen Beschleunigungssprünge werden vom Fahrer als unangenehmes „Rucken“ wahr-



genommen. Deshalb wurde eine Steuerung des Motormoments während der Schaltungsphasen realisiert, siehe Abschnitt 6.1.3. In der Abb. 10.2 sind die Verläufe aus der Simulation mit Motormomentbegrenzung dargestellt. Die Verläufe der Simulationsgrößen Geschwindigkeit, Drosselklappenstellung, Bremspedalstellung, gewählter Gang und Motordrehzahl entsprechen dem aus der Literatur zu erwartenden Verlauf. Aus den Verläufen der normierten Drosselklappen- und Bremspedalstellung kann das Verhalten des Fahrers beurteilt werden. Dabei entsprechen ruhige Verläufe einer guten Reglereinstellung des Longitudinal-Fahrers. Die Gänge werden bei einer Drosselklappenstellung von ca.  $\varphi_{DK} = 0.5$  und bis zu einer Motordrehzahl von ca.  $\omega_M = 4200$  ausgedreht.

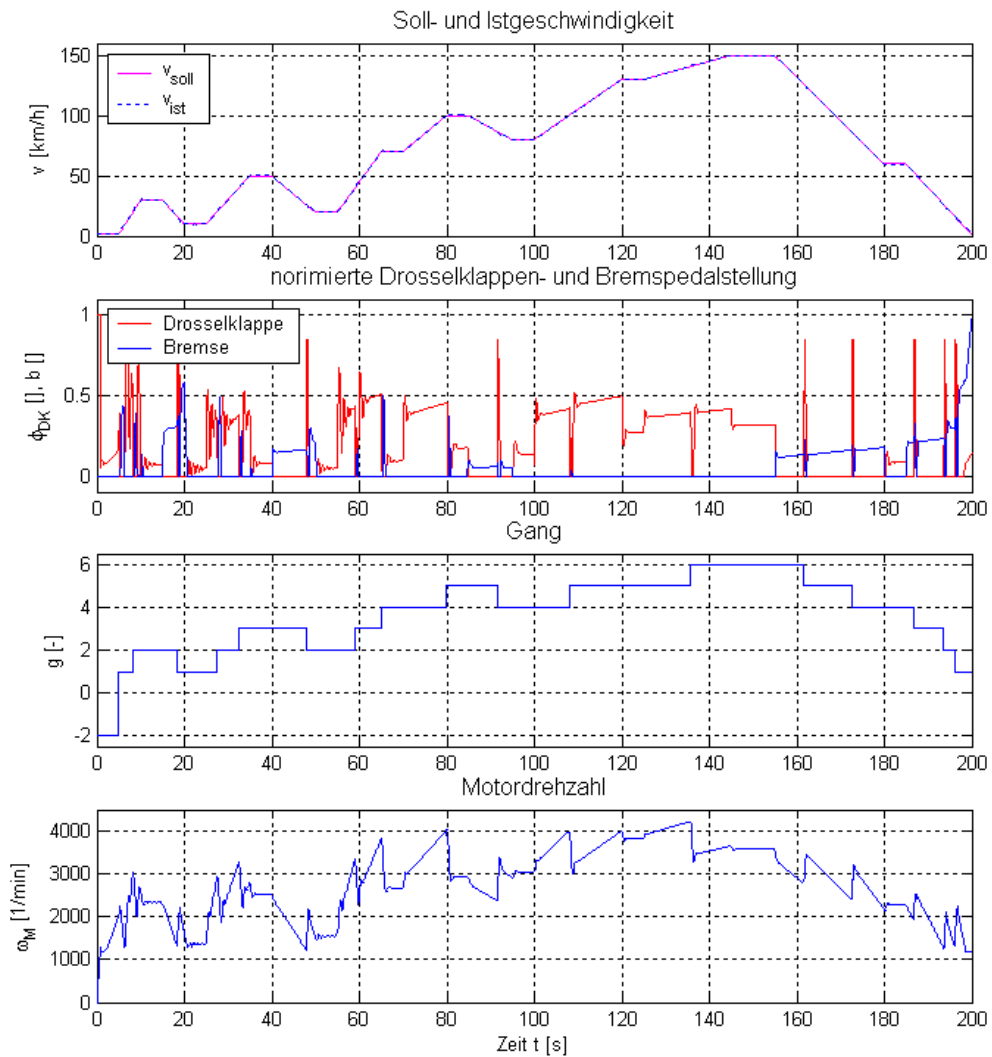


Abbildung 10.2: Ausgewählte Offline-Simulationsergebnisse des Längsfahrzyklus mit Steuerung des Motormoments

In der Abb. 10.3 sind die Verläufe der Simulationsgrößen Geschwindigkeitsfehler, Fahr-

zeuglängsbeschleunigung und Radschlupf dargestellt. Es ist zu erkennen, dass die Beschleunigungs- und Radschlupfspitzen geringer ausfallen. Außerdem ist der Geschwindigkeitsfehler geringer. Aus diesen Betrachtungen heraus werden im weiteren alle Simulationen mit Steuerung des Motormoments während der Schaltung durchgeführt.

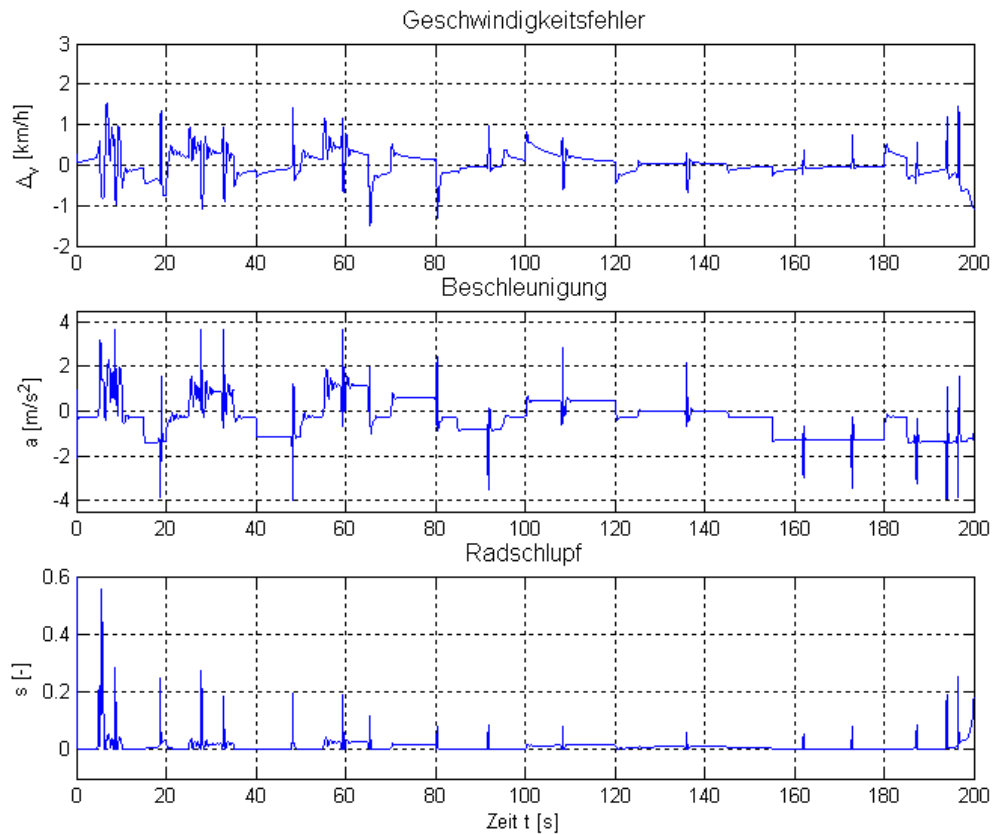


Abbildung 10.3: Geschwindigkeitsfehler, Fahrzeuglängsbeschleunigung und Radschlupf des Längsfahrzyklus mit Steuerung des Motormoments

### Stationäre Kreisfahrt

Mit dem Manöver Kreisfahrt wird die Querdynamik des Fahrzeugmodells untersucht. In der Abb. 10.4 sind ausgewählte Simulationsergebnisse dargestellt. Nachdem das Fahrzeug die Geschwindigkeit  $v_{Fx} = 50 \text{ km/h}$  erreicht hat, beginnt die Kurvenfahrt. Dazu wird der Lenkwinkel in 5 s kontinuierlich auf  $90^\circ$  erhöht und für die restliche Simulationszeit konstant gehalten.

Die Verläufe der Simulationsergebnisse Gierwinkelgeschwindigkeit, Querbewegung und Schwimmwinkel zeigen die zu erwartenden Ergebnisse. Die Querbewegung und die Gierwinkelgeschwindigkeit geben recht guten Aufschluss darüber, wie das Fahrzeug auf den Lenkwinkelverlauf reagiert. Bei größeren Lenkwinkeln ist aus physikalischer Sicht auch ein Anstieg der Querbewegung und der Gierwinkelgeschwindigkeit zu erwarten. Eine weitere

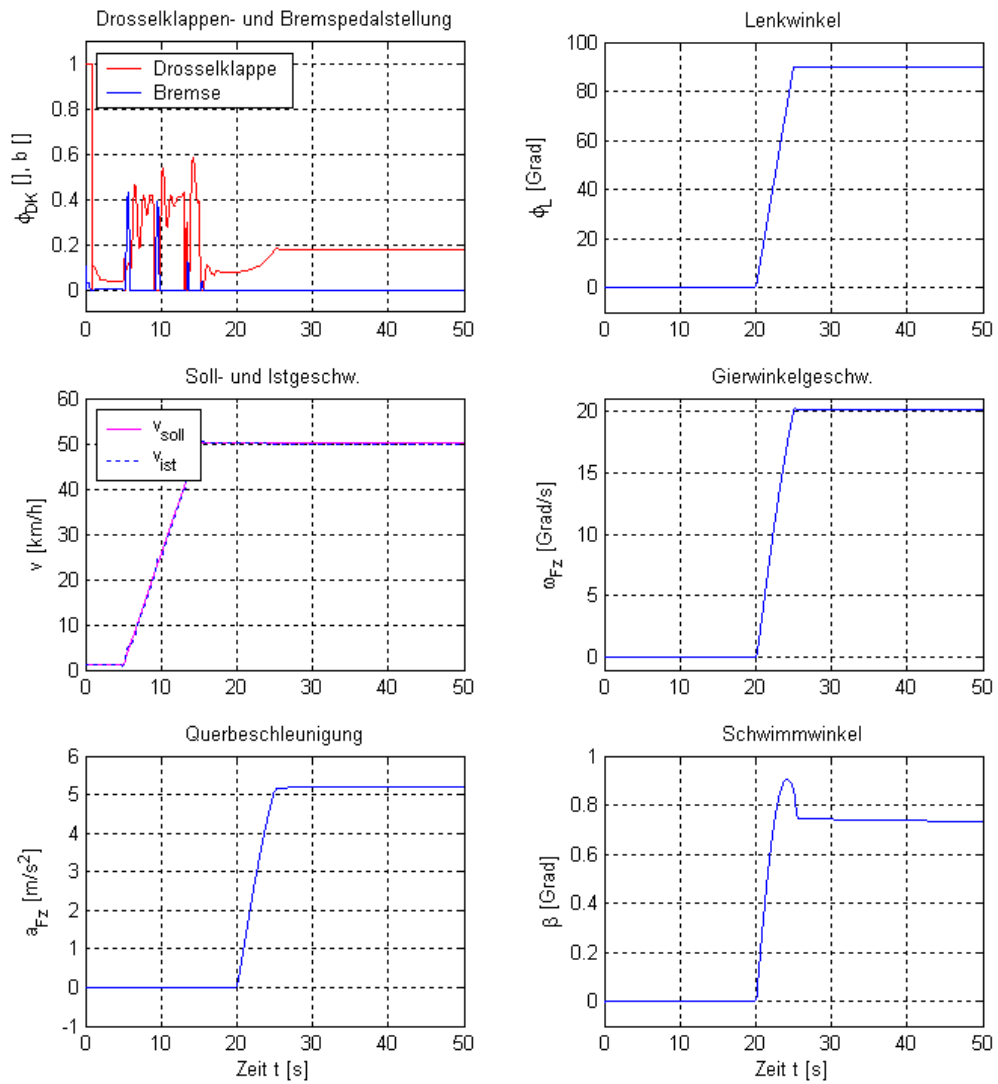


Abbildung 10.4: Ausgewählte Simulationsergebnisse der stationären Kreisfahrt

wichtige Größe zur Beurteilung der Fahrstabilität ist der Schwimmwinkel  $\beta$ .

## 10.2 Echtzeitsimulation

Nach der Beschreibung der Offline-Verifizierung des objektorientierten Fahrzeugmodells wird im Folgenden die Entwicklungsumgebung zur Echtzeitsimulation und Ergebnisdarstellung des Modells vorgestellt. Weil zurzeit keine direkte Echtzeitsimulation mit *Dymola* möglich ist, wird dafür das Simulationssystem *xPC-Target* von *The Mathworks* eingesetzt. Das *xPC-Target*-System erlaubt die Echtzeitsimulation eines *Simulink*-Modells auf einer separaten echtzeitfähigen Hardware. Als Echtzeitrechner wird ein handelsüblicher Personalcomputer mit einem *Intel-Pentium*-Prozessor und einer Taktfrequenz von 2.66 GHz eingesetzt. Alle Fahrzeugkomponenten sind in Software realisiert, wodurch keine Hardware zum Anschluss von realen Bauteilen (z. B. EGS) benötigt wird. Die Kommunikation mit dem Echtzeitsystem erfolgt über ein Ethernet-Netzwerk. Bei der Echtzeitsimulation über das Institutsnetzwerk traten während der Datenübertragung Fehler auf, dies konnte mit einer Netzwerkdirektverbindung behoben werden.

Das *xPC-Target*-System bietet selbstverständlich auch die Möglichkeit mit anderen Echtzeitrechensystemen wie beispielsweise Power-PCs und Signalprozessorsystemen zu arbeiten. Weiter stehen in der Bibliothek von *xPC-Target* verschiedene Blöcke zur Anbindung von E/A-Karten zur Verfügung.

### 10.2.1 Entwicklungsumgebung

Mit *Matlab/Simulink* und *xPC Target* ist es möglich, auf Standard-PCs Echtzeitanwendungen zu entwickeln und zu testen. Dabei läuft auf dem „Host-PC“ die Software zur Erstellung und Steuerung der Echtzeitsimulation. Auf einem „Target-PC“ mit dem echtzeitfähigen Betriebssystem von *xPC Target* läuft allein die Echtzeitanwendung. Der Entwicklungsprozess für die Echtzeitsimulation des in *Dymola* erstellten Modells kann in drei Phasen untergliedert werden:

1. Design und Offline-Simulation mit *Simulink*
2. Code-Generierung und Kompilierung
3. Online-Simulation.

In der ersten Phase wird über den in *Simulink* enthaltenen *Dymola*-Block das *Dymola*-Modell importiert. Der mit *Dymola* erstellte C-Code des Modells wird über die *Dymola-Simulink*-Schnittstelle in eine „Simulink S-Function“ umgewandelt. Vor der Importierung ist

es jedoch notwendig, die Ein- und Ausgänge des *Dymola*-Modells in der obersten Modellebene zu definieren. Zusätzlich müssen die Parameter, die während der Echtzeitsimulation zur Verfügung stehen sollen, definiert werden. Nachdem der *Dymola*-Block mit den entsprechenden Ein- und Ausgängen in *Simulink* zur Verfügung steht, kann das Modell mit weiteren *Simulink*-Blöcken vervollständigt werden. Weiter müssen die Signale, die bei der Echtzeitsimulation angezeigt und gespeichert werden, festgelegt werden. Zum Schluss kann in einer Offline-Simulation das Modell untersucht werden. Eine Beschreibung der Anpassung des Modells an die speziellen Erfordernisse der Echtzeitsimulation erfolgt in Abschnitt 10.2.2.

Nachdem das Modell in *Simulink* zur Verfügung steht, kann mit dem Real-Time Workshop echtzeitfähiger Code erstellt werden. Dieser Code wird über die Netzwerkverbindung an den Echtzeitrechner (Target-PC) gesendet. Damit kann die eigentliche Simulation beginnen.

Die Steuerung der Simulation erfolgt im Wesentlichen über den „Host-PC“ und kann mittels Kommandoeingabe in *Matlab*, über Skripte oder komfortabel über das „Remote Control Tool“ von *xPC Target* erfolgen. Des Weiteren können Ausgangssignale und Parameter online am Bildschirm des „Target-PCs“ oder am „Host-PC“ verfolgt werden. Die Online-Darstellung benötigt jedoch Rechenzeit, weshalb wird im Folgenden keine Onlinedarstellung am „Target-PC“ erfolgt. Nach der Echtzeitsimulation können die gespeicherten Ausgangssignale auf den „Host-PC“ übertragen und in *Matlab* bearbeitet werden. Im Anhang A ist ein Skript zur Speicherung der Echtzeitsimulationsergebnisse dargestellt.

### 10.2.2 Anpassung der Simulationseinstellungen

Um eine Echtzeitsimulation des Fahrzeugmodells mit geringer Rechenzeit zu ermöglichen, müssen schon in *Dymola* einige Simulationsparameter vorteilhaft gewählt werden. Der *Dymola*-Integrator ist gegenüber dem von *Simulink* verwendeten Integrator besser für die Echtzeitanwendungen geeignet. Mit der Option „Inline Integration“ wird von *Dymola* ein C-Code erstellt, der den *Dymola*-Integrationsalgorithmus „Explizit Euler“ enthält. So wird auch bei der Echtzeitsimulation der *Dymola*-Integrationsalgorithmus verwendet. Weiter existieren in *Dymola* versteckte Schalter, mit denen die Erstellung des C-Codes beeinflusst werden kann. Im Anhang B ist ein Skript dargestellt, mit dem der optimale Echtzeitcode erstellt werden kann.

Nachdem das Fahrzeugmodell in *Simulink* modelliert wurde, müssen noch die Simulationsparameter für die Erstellung des Echtzeitcodes angepasst werden. Für die Simulation wird der „Euler“-Algorithmus mit der festen Schrittweite  $t = 1 \text{ ms}$  verwendet. Außerdem müssen für den *Realtime Workshop* noch entsprechende Einstellungen angepasst werden. In der Tabelle 10.1 sind die angepassten Simulationseinstellungen dargestellt.

Nachdem die Simulationseinstellungen vorgenommen wurden, kann das Fahrzeugmodell

Simulationsparameter	Einstellung	Beschreibung
System target file	xpctarget.tlc	Codegeneration für das xPC-System
Execution mode	Real-Time	Ausführungsmodus mit Echtzeitbedingung, d. h. bei Überschreiten der Schrittweite wird die Simulation abgebrochen
Signal Logging buffer size in doubles	6500000	Speicherreservierung für die Signale

Tabelle 10.1: *Real-Time-Workshop*-Einstellungen

kompiliert werden. Nach erfolgreicher Erstellung wird die Echtzeit-Anwendung auf den „Target-PC“ heruntergeladen und steht für die Simulation bereit.

### 10.2.3 Ausgewählte Simulationsergebnisse

Im Folgenden werden die Echtzeitsimulationen des detaillierten Fahrzeugmodells verifiziert und ausgewählte Ergebnisse dargestellt. Aus Gründen des Platzes werden dafür exemplarisch zwei Fahrmanöver ausgewählt. Für die Verifizierung werden die dargestellten Ergebnisse mit zu erwartenden Ergebnissen aus der Literatur (z. B. [Hol01], [Mit90] und [Mit95]) überprüft. Durch Vergleich der Simulationsergebnisse von der Offline- und Echtzeitsimulation wird der numerische Fehler untersucht.

#### Längsfahrzyklus

Zur Verifizierung der Längsdynamik des Fahrzeugmodells werden die Ergebnisse der Echtzeitsimulation mit denen der Offline-Simulation, siehe Abschnitt 10.1.3, verglichen. Dabei wird für eine Simulationsdauer von  $t = 100$  s das gleiche Geschwindigkeitsprofil abgefahren. In der Abb. 10.5 sind die Verläufe der Simulationsgrößen Fahrzeuggeschwindigkeit, Drosselklappenstellung, Bremspedalstellung, eingelegter Gang und Motordrehzahl dargestellt. Die Diagramme zeigen jeweils einen Vergleich zwischen den in *Dymola* (blau) und den in Echtzeit (rot) simulierten Verläufen repräsentativer Fahrzeugzustandsgrößen. In allen Diagrammen ist eine gute Übereinstimmung der beiden Verläufe zu sehen. Die numerischen Fehler der verifizierten Offline- und Echtzeitsimulation stimmen überein. Damit ist für die Echtzeitsimulation numerische Stabilität gegeben.

#### $\mu$ -split-Bremmung

Für die Untersuchung des Systems Rad-Straße wurde eine  $\mu$ -split-Bremmung simuliert. Das Fahrzeug fährt mit einer Geschwindigkeit von  $v = 50$  km/h auf einer normalen Straße (gleiche Reibwerte). Bei  $t = 20$  s beginnt der Bremsvorgang und bei  $t = 20.5$  s kommt das Fahrzeug

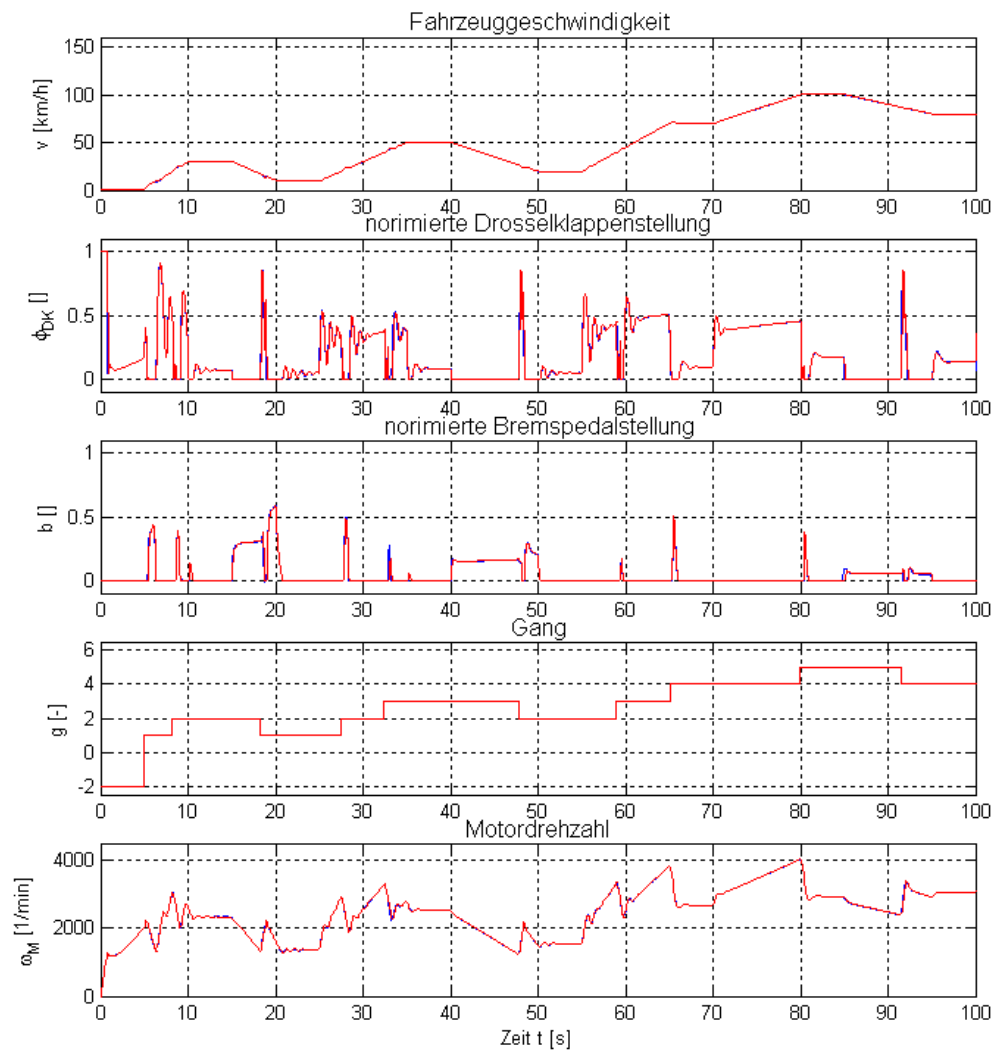


Abbildung 10.5: Vergleich der Simulationsergebnisse des Längsfahrzyklus

auf die  $\mu$ -split-Fahrbahn. Die Räder 1 und 2 befinden sich dabei auf der glatten Fahrbahnseite.

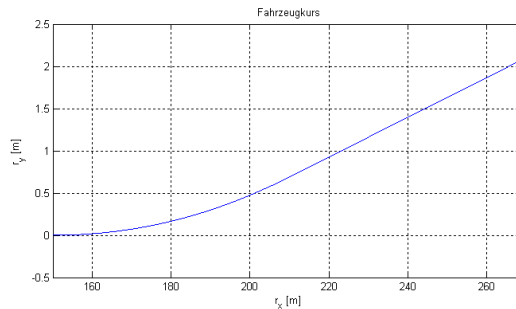


Abbildung 10.6: Fahrzeugkurs der  $\mu$ -split-Bremung

In der Abb. 10.6 ist der simulierte Fahrzeugkurs dargestellt. Entgegen der Realität bewegt sich das Fahrzeug auf die glatte Fahrbahnseite. In der Abb. 10.7 sind die simulierten Fahrzeuggrößen Bremspedalstellung, Fahrzeuggeschwindigkeit, Fahrzeugquerbeschleunigung, Gierwinkelgeschwindigkeit sowie Schlupf und Schlupfwinkel an den gebremsten Vorderrädern dargestellt. Im Diagramm des Radschlupfs der Vorderräder ist wie erwartet zu erkennen, dass das Rad 1 einen größeren Schlupf als Rad 2 aufweist. Nach Untersuchung der Simulationsergebnisse wurde festgestellt, dass der Vorspurwinkel der Vorderräder zu groß parametrisiert wurde. Auf einer  $\mu$ -split-Fahrbahn ist das durch die Vorspur erzeugte negative Moment von Rad 1 (glatte Fahrbahn) kleiner als das positive Moment von Rad 2 (trockene Fahrbahn). Bei dem Bremsmanöver wird deshalb das von der Bremse erzeugte negative Giermoment durch das Moment der Vorspur überkompensiert und das Fahrzeug bewegt sich weiter auf die glatte Fahrbahn. Nachdem sich alle Räder auf der gleichen Fahrbahnhälfte befinden, tritt kein Giermoment mehr auf und das Fahrzeug fährt geradeaus weiter. Die Parametrisierung der Vorspur und des Radsturzes erfolgt über komplexe nichtlineare Kennlinien. Weil eine exakte Abbildung der Fahrzeugzustandsgrößen nicht im Vordergrund dieser Arbeit stand, wurde diese inkorrekte Parametrisierung beibehalten.

### Dynamische Kurvenfahrt

Mit einer beschleunigten Kurvenfahrt wird die Fahrzeugquerdynamik des Echtzeitmodells aufgezeigt. In der Abb. 10.8 sind die Verläufe der Simulationsgrößen Drosselklappenstellung, Bremspedalstellung, Lenkwinkel, Fahrzeuglängsgeschwindigkeit, eingelegter Gang, Fahrzeugquerbeschleunigung und Gierwinkelgeschwindigkeit dargestellt. Das Fahrzeug wird auf eine Geschwindigkeit von  $v = 50$  km/h beschleunigt. Bei  $t = 20$  s beginnt die Kurvenfahrt, dazu wird in 3 Sekunden der Lenkwinkel kontinuierlich bis auf  $\varphi = 60^\circ$  erhöht. Nach 9 Sekunden wird der Lenkwinkel wieder kontinuierlich auf  $\varphi = 0^\circ$  zurückgestellt. Während dieser



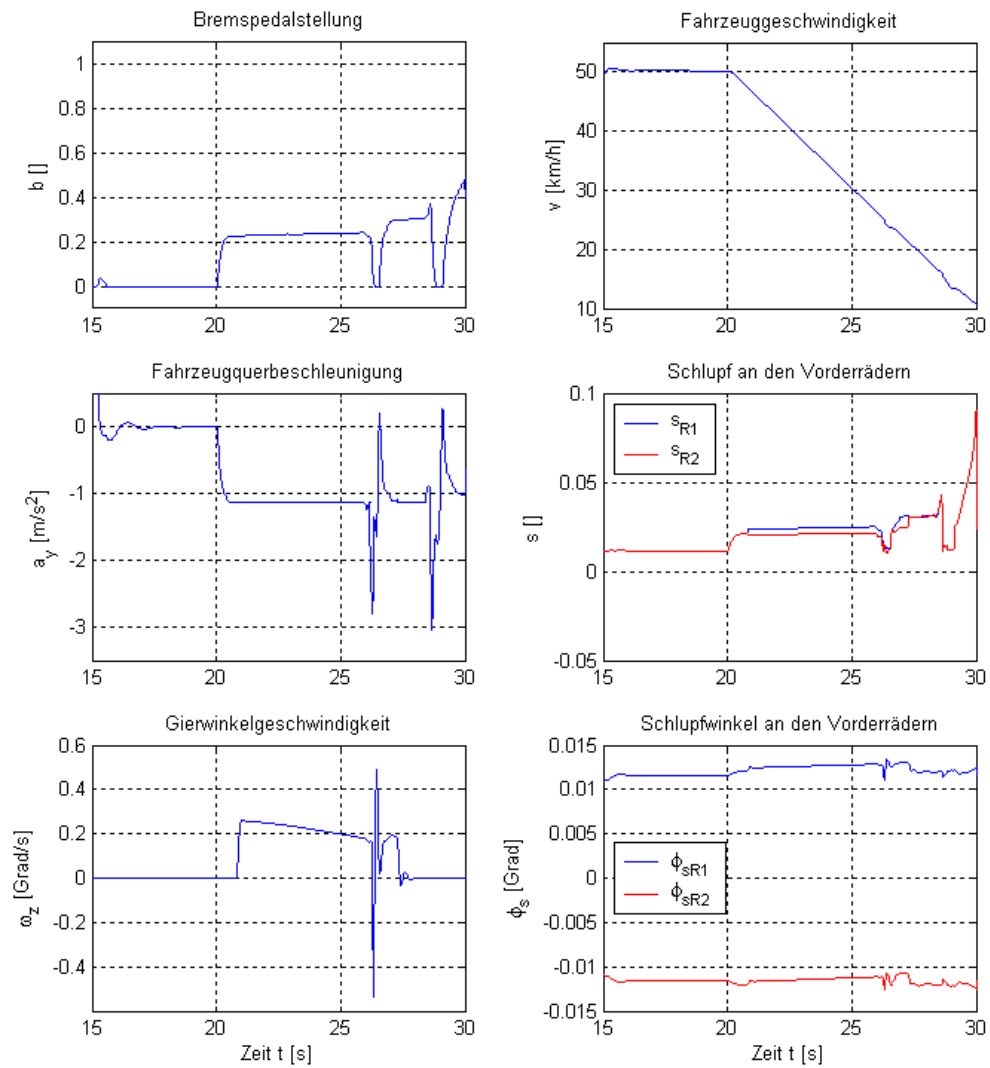


Abbildung 10.7: Ausgewählte Simulationsergebnisse der  $\mu$ -split-Bremmung

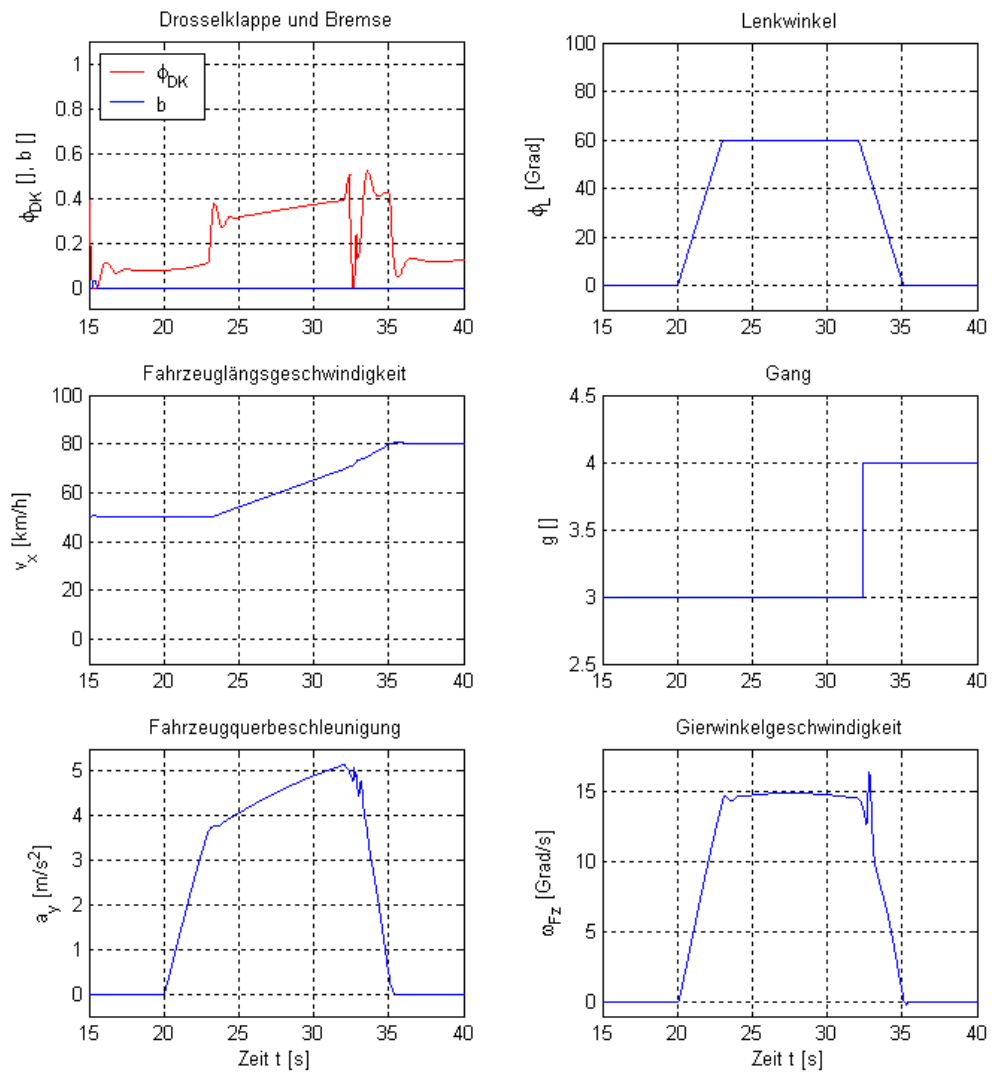


Abbildung 10.8: Ausgewählte Simulationsergebnisse der dynamischen Kurvenfahrt

Kurvenfahrt wird das Fahrzeug auf eine Geschwindigkeit von  $v = 80 \text{ km/h}$  beschleunigt. Der Verlauf der Fahrzeugquerbeschleunigung weist beim Einlenken, positive Beschleunigung auf. Bei einer Fahrzeuglängsbeschleunigung in der Kurve nimmt die Querbeschleunigung weiter zu. Weiter ist beim Gangwechsel wegen der Momente der Hinterräder eine Schwankung der Querbeschleunigung zu beobachten. Ein Ausbrechen des Heck erfolgt aber bei dieser Geschwindigkeit nicht. Der simulierte Fahrzeugkurs, siehe Abb. 10.9, zeigt den erwarteten Verlauf.

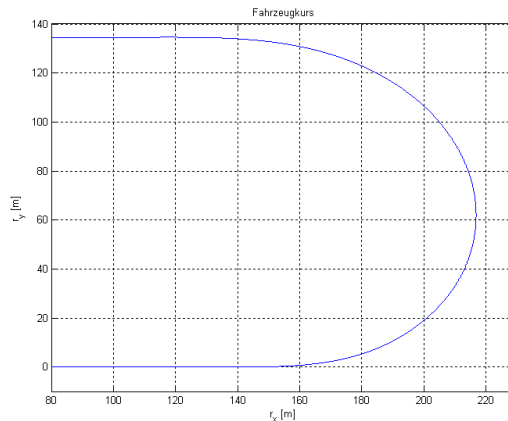


Abbildung 10.9: Fahrzeugkurs der dynamischen Kurvenfahrt

Außerdem sind die Verläufe der Radschlupfe und Schlupfwinkel für das Fahrverhalten von Interesse. Deshalb ist in der Abb. 10.10 der Verlauf des Schlupfs und der Schlupfwinkel der einzelnen Räder dargestellt. Bei der beschleunigten Kurvenfahrt ist ein gutmütiges Fahrverhalten des Fahrzeugs zu erkennen, d. h. das Fahrzeug schiebt leicht über die Vorderräder. Die unterschiedlichen Schlupfwinkel der Vorderräder sind durch die Vorspur und die Lenkung bedingt, siehe Abschnitt 7.2.2. Bei ca. 32 Sekunden erfolgt ein Gangwechsel, dabei ist eine Schlupfspitze an den Hinterrädern zu erkennen. Außerdem ist ebenfalls eine Zunahme der Gierwinkelgeschwindigkeit in der Abb. 10.8 zu beobachten. Mit diesem Fahrzeugmodell können somit Effekte des Antriebsstrangs auf die Fahrdynamik simuliert werden.

### 10.3 Fahrzeugmodell mit 3D-Effekten im Antriebsstrang

Für die Simulation des Fahrzeugmodells mit 3D-Effekten im Antriebsstrang wurde wegen der neuen Komponenten die separate Bibliothek „PowerVehicle3D“ mit allen notwendigen Komponenten erstellt. In der Abb. 10.11 ist die oberste Ebene des Fahrzeugmodells dargestellt.

Untersuchungen haben gezeigt, dass das Fahrzeugmodell ohne 3D-Effekte im Antriebsstrang mit der „VehicleDynamics“-Bibliothek Version 0.8 eine langsamere Rechengeschwin-

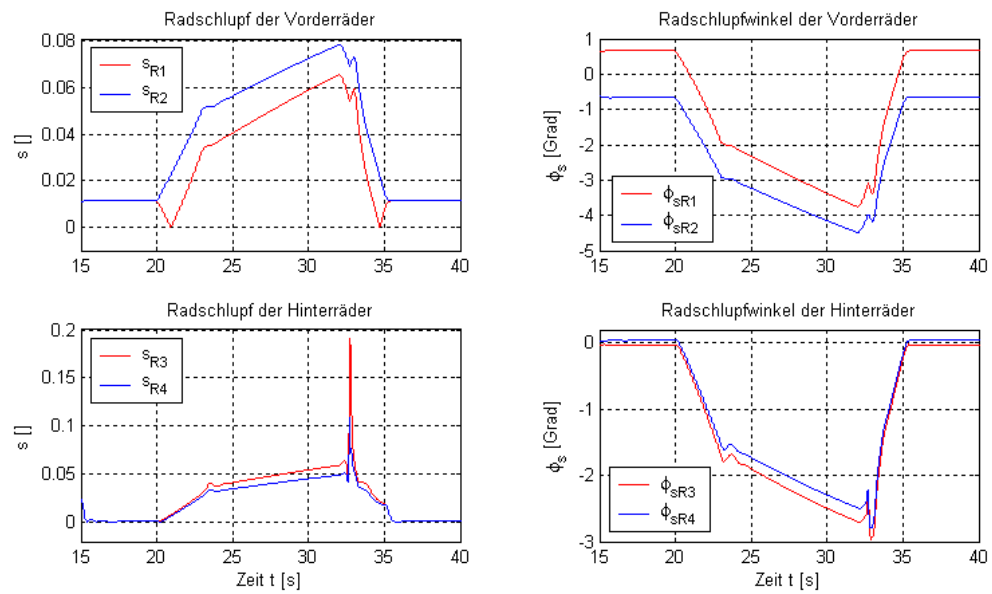


Abbildung 10.10: Schlupf und Schlupfwinkel der einzelnen Räder

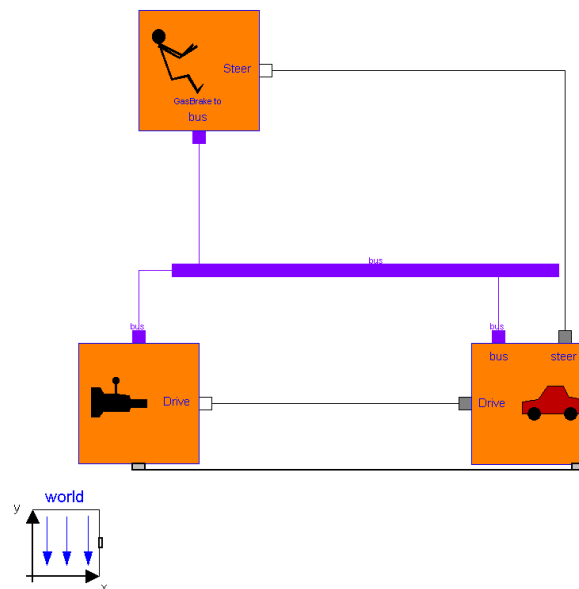


Abbildung 10.11: Objektdiagramm des Fahrzeugmodells mit 3D-Effekten im Antriebsstrang

digkeit aufweist. Dies ist auf die neuen Komponenten der „MultiBody“-Bibliothek Version 1.0 zurückzuführen. Insbesondere haben sich die mechanischen 3D-Schnittstellen verändert, wodurch der optimierte Simulationscode noch nicht die hohe Effizienz wie der Simulationscode des oben beschriebenen Fahrzeugmodells (siehe Abschnitt 9.2) erreicht. Es ist daher nicht möglich eine Echtzeitsimulation durchzuführen. Deshalb wird das Fahrzeugmodell in einer Offline-Simulation mit *Dymola* untersucht.

### 10.3.1 Vergleich der Simulationsergebnisse

Das Fahrzeugmodell mit 3D-Effekten im Antriebsstrang wird für den Vergleich mit dem Integrationsalgorithmus „dassl“ und einem maximalen Fehler von 0.0001 simuliert. Für den Vergleich der Simulationsergebnisse wird die dynamische Kurvenfahrt betrachtet. In den nachfolgenden Abbildungen werden die Verläufe des Fahrzeugmodells ohne 3D-Effekte (blau) und mit 3D-Effekte (rot) abgebildet.

In der Abb. 10.12 sind die Verläufe der Simulationsgrößen Drosselklappenstellung, Lenkwinkel, Fahrzeuggeschwindigkeit, eingelegter Gang, Fahrzeuglängsbeschleunigung und Fahrzeugquerbeschleunigung dargestellt. Die Fahrzeuge werden mit einer Geschwindigkeit von  $v_x = 10$  km/h initialisiert und beschleunigen auf eine Geschwindigkeit von  $v_x = 50$  km/h vor der Kurvenfahrt. Durch kleine Unterschiede bei der Drosselklappenstellung und der Fahrzeuglängsgeschwindigkeit kommt es bei der Simulation des Fahrzeugs mit 3D-Effekten zu zeitigeren Gangwechseln. Ab der Zeit  $t = 20$  s beginnt die Kurvenfahrt entsprechend dem Lenkwinkelverlauf. Bei den Verläufen Geschwindigkeit, Längs- und Querb beschleunigung treten keine großen Abweichungen zwischen den Simulationsergebnissen auf.

Die Verläufe der Fahrzeugkurse, siehe Abb. 10.13, zeigen keinen großen Unterschied. Hierbei ist zu beachten, dass der Fahrzeugkurs über den vorgegebenen Lenkwinkel nur gesteuert wird und somit Abweichungen nicht ausgeregelt werden.

In der Abb. 10.14 sind die Verläufe der Momente und Winkelgeschwindigkeiten um die Achsen (x,y,z) des fahrzeugfesten Koordinatensystems dargestellt. Bei den Verläufen ohne 3D-Effekte im Antriebsstrang sind keine Momente auf das Fahrzeug zu beobachten. Dem gegenüber sind deutliche Momente auf das Fahrzeug mit 3D-Effekte im Antriebsstrang festzustellen. Diese Momente bewirken ein deutliches Wanken (um x-Achse) und Nicken (um y-Achse) des Fahrzeugaufbaus. Bei der Kurvenfahrt treten jedoch keine großen Gierwinkelgeschwindigkeiten auf, die ein Ausbrechen des Fahrzeugs darstellen würden. Bei normalen Fahrmanövern haben die Momente vom Antriebsstrang auf die Fahrdynamik des Fahrzeugs eine geringe Wirkung. Das Fahrzeugmodell kann für Komfortbetrachtungen genutzt werden.

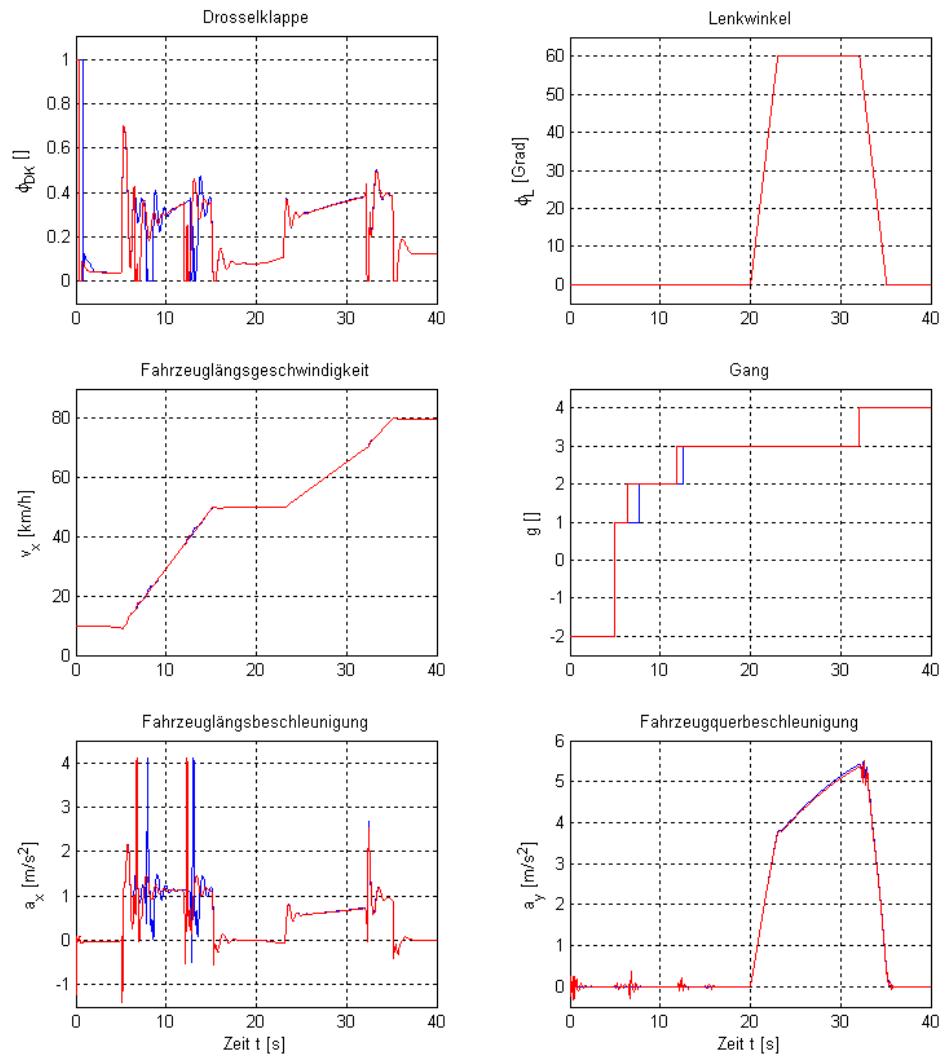


Abbildung 10.12: Vergleich ausgewählter Simulationsergebnisse ohne und mit 3D-Effekten im Antriebsstrang

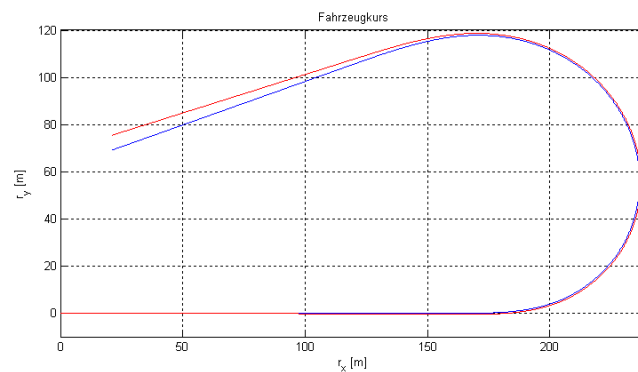


Abbildung 10.13: Vergleich des Fahrzeugkurses ohne und mit 3D-Effekten im Antriebsstrang

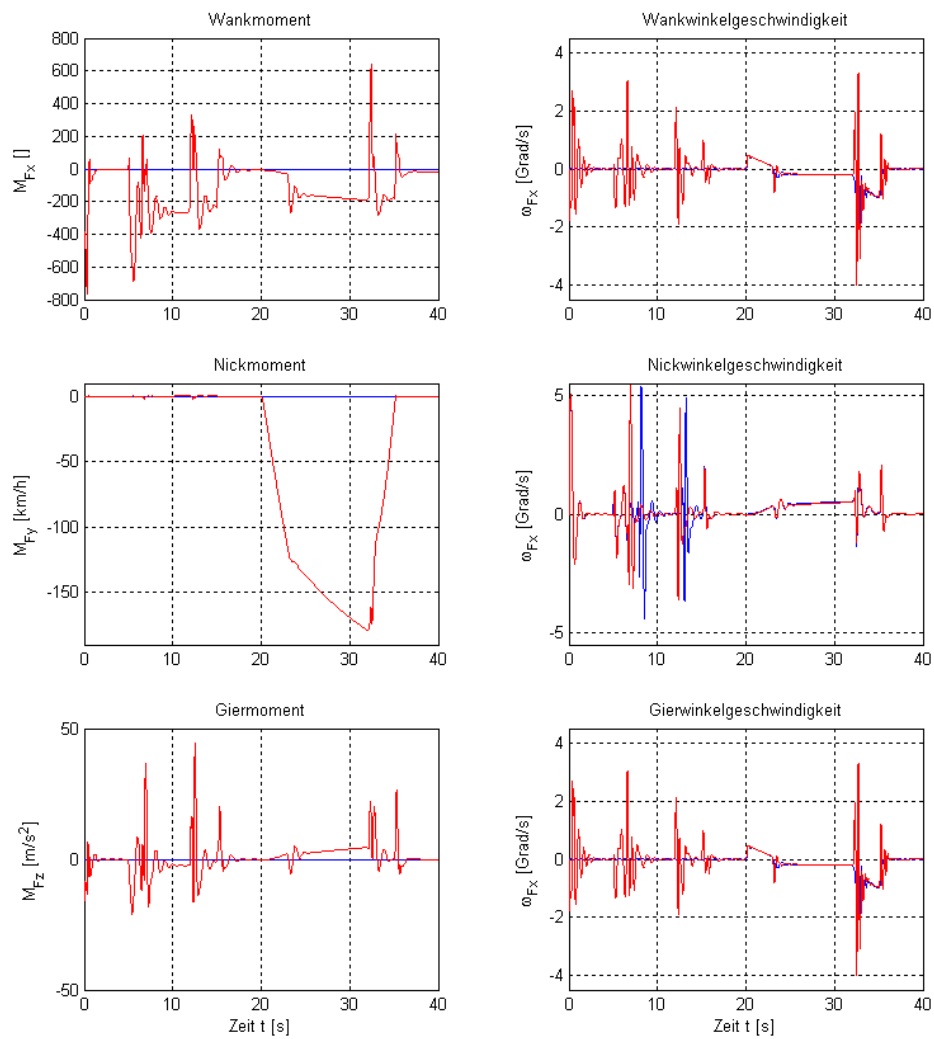


Abbildung 10.14: Vergleich der Momente und der Winkelgeschwindigkeiten ohne und mit 3D-Effekten im Antriebsstrang

# 11 Wesentliche Ergebnisse und Schlussfolgerungen

Bei der Modellierung von detaillierten Fahrzeugen für die Simulation von Fahr- und Antriebsstrangdynamik müssen vielfältige Probleme gelöst werden, siehe Kapitel 2. Das Simulationssystem *Modelica* mit *Dymola* bietet dafür eine gute Lösung. Die objektorientierte Struktur von *Modelica* erlaubt einen transparenten Aufbau der Modelle, und mit den vorhandenen *Modelica*-Bibliotheken können einfach komplexe Modelle entworfen werden. Für die Modellierung werden jedoch fundierte Kenntnisse in den entsprechenden Teilgebieten der Fahr- und Antriebsstrangdynamik vorausgesetzt. Durch den komponentenweisen Aufbau der Modelle können entsprechende Fahrzeugkomponente leicht ausgetauscht werden. Dies bringt bei der Erstellung von unterschiedlichen Fahrzeugmodellen große Vorteile. Außerdem bietet *Dymola* mit seiner grafischen Oberfläche Neueinsteigern die Möglichkeit, sich schnell einzuarbeiten.

Die Verbindung von *Dymola* und *xPC-Target* eignet sich besonders zur Echtzeitsimulation von detaillierten, objektorientierten und strukturvariablen Modellen. *Dymola* bietet gegenüber *Matlab/Simulink* den Vorteil des schnelleren und numerisch stabileren Simulierens von Modellen. Des Weiteren ermöglicht *xPC Target* eine kostengünstige Echtzeitsimulation mit standardisierter Hardware.

Die Parametrisierung und die Fehlersuche gestalteten sich in einigen Fällen sehr schwierig, weil die Fehlermeldungen detaillierte Kenntnisse von der Struktur des Modells voraussetzen. Für das detaillierte Fahrzeugmodell sind diese Probleme bei der Simulation nicht trivial. Denn bei den verwendeten Bibliotheken „VehicleDynamics“ und „PowerTrain“ wurde für die Echtzeitsimulation eine zum Teil ungünstige Parametrisierung der Komponenten festgestellt. In dieser Arbeit bestand die Aufgabe, das erstellte Fahrzeugmodell so zu parametrisieren und zu initialisieren, dass es den Echtzeitbedingungen genügt. Am DLR lagen nicht alle Parameter (mehr als 10000) für ein spezifisches Fahrzeugmodell vor. So wurden die meisten Parameter der Komponenten aus den entsprechenden Bibliotheken übernommen. Dies führte bei der Simulation des Fahrzeugmodells zu teilweise von der Realität abweichenden Ergebnissen.

In *Modelica* können sehr einfach Anfangswertgleichungen erstellt werden. Diese Gleichungen ermöglichen die Behandlung von schwierigen Initialisierungsproblemen. Für die Initiali-



sierung einer konstanten Anfangsgeschwindigkeit des Fahrzeugs können für die Zustandsableitungen Anfangswerte festgelegt werden. Außerdem können strukturvariable Systeme einfach initialisiert werden. Mit dem bei der Übersetzung und Simulation des Modells erstellten Log-Files ist es in *Dymola* einfach, das detaillierte Fahrzeugmodell zu initialisieren.

Das detaillierte Fahrzeugmodell enthält viele strukturvariable Komponenten, wie z. B. Bremsen, Kupplungen oder Räder. Diese Komponenten bringen jedoch bei der Echtzeitsimulation Probleme. Denn bei der Simulation kommt es zu mehreren Zustandsumschaltungen in kürzester Zeit und die Rechenzeit für einen Rechenschritt wird zu groß, d. h. die Echtzeitbedingung wird nicht mehr erfüllt. Unter Beachtung der Schrittweite von  $t = 1$  ms muss das Fahrzeugmodell vereinfacht werden, oder man setzt spezielle, schnellere Echtzeitrechnersysteme für die Echtzeitsimulation ein. In dieser Arbeit wurde deshalb ein vereinfachtes Fahrzeugmodell mit zwei Bremsen an der Vorderachse realisiert. Mit diesem Modell wurden die Fahr- und Antriebsstrangdynamik mittels Echtzeitsimulation untersucht.

Mit dem Modell lassen sich in verschiedenen Fahrmanövern alle relevanten Fahrzustandsgrößen untersuchen. Mit der Simulation des Längsfahrzyklus wurde gezeigt, wie die Größen Fahrzeugbeschleunigung und Schlupf an den angetriebenen Rädern von der Drosselklappenstellung, der Bremspedalstellung und dem Schaltzustand abhängen. Bei der Simulation wurden deutliche Sprünge in der Fahrzeugbeschleunigung festgestellt, welche vom Fahrer als unangenehmes Rucken empfunden werden. Eine Verbesserung des Fahrkomforts beim Schalten konnte durch eine Steuerung des Motormoments während der Schaltphase erreicht werden. Bei dem Fahrmanöver der  $\mu$ -split-Beschleunigung ergaben sich, aufgrund ungünstiger Parametrisierung der Vorspur, von der Realität abweichende Simulationsergebnisse. Mit einer verbesserten Parametrisierung der Kennlinie für den Vorspurwinkel kann jedoch das richtige Fahrverhalten simuliert werden. Bei den Manövern der Kurvenfahrt wurde die Querdynamik des Fahrzeugmodells untersucht.

Darüber hinaus können mit *Modelica* die Momente, die der Antriebsstrang auf die Karosserie überträgt, untersucht werden. Bei dem Vergleich einer Kurvenfahrt mit und ohne 3D-Effekten im Antriebsstrang wurde gezeigt, dass sich die 3D-Effekte im Antriebsstrang nur gering auf die Fahrdynamik auswirken. Die Bewegung der Karosserie infolge der 3D-Effekte des Antriebsstrangs sind jedoch für Komfortbetrachtungen von Interesse. Dieses Modell enthält außerdem vier Bremsen, weshalb es für zukünftige Untersuchungen in fahrdynamischen Grenzbereichen interessant ist.

Für die Entwicklung von Fahrdynamikregelungen kann das detaillierte Fahrzeugmodell mit vier Bremsen aus der erstellten „PowerVehicle“-Bibliothek verwendet werden. Mit dem Integrationsverfahren „Euler“ und einer Schrittweite von  $t = 0.5$  ms konnten mit diesem Modell Offline-Simulationen bis in fahrdynamische Grenzsituationen durchgeführt werden. Dabei

stellt das Fahrzeugmodell alle notwendigen Größen für die Entwicklung von Antriebsstrang-Management-Systemen und Fahrdynamikregelungen zur Verfügung.

## 12 Zusammenfassung und Ausblick

An neue Generationen von Automobilen werden ständig wachsende Anforderungen hinsichtlich Fahrkomfort, Agilität und Fahrsicherheit gestellt. Insbesondere kommt dem Einsatz von elektronischen Steuergeräten und deren Integration in das gesamte Fahrzeug eine zentrale Rolle zu. Dies ermöglicht weitere Innovationen für die Steuerungs- und Regelungstechnik im Automobilbereich, jedoch erhöht sich wegen der gestiegenen Komplexität der Entwicklungsaufwand. Hierzu stellt die Simulationstechnik ein mächtiges Entwicklungs- und Testwerkzeug dar, das kosten- und zeitintensive Versuche am realen Fahrzeug teilweise ersetzen kann [Hol01]. Darüber hinaus kann durch Simulation das Verständnis für komplexe Zusammenhänge im Fahrzeug verdeutlicht und nur schwer zu messende Fahrzeuggrößen betrachtet werden.

Wie aus dem Stand der Technik hervorgeht, wurden für das komplette Fahrzeug regelungstechnische Modelle eingesetzt, die die Fahr- und Antriebsstrangdynamik in die Teilgebiete Längs-, Vertikal- und Horizontal-dynamik sowie Lenkung und Rad-Straße aufteilen. Zum Testen und Optimieren von komplexen Fahrzeugregelungen müssen alle Teilgebiete betrachtet werden, wodurch der Entwicklungsaufwand für detaillierte Fahrzeugmodelle ansteigt. Darüber hinaus ist eine Echtzeitsimulation des Fahrzeugmodells nur mit einer erneuten Idealisierung oder großen Rechenanlagen möglich.

Zur Verringerung des Entwicklungsaufwands für detaillierte Fahrzeugmodelle wurde in dieser Arbeit der objektorientierte Modellierungsansatz von *Modelica* untersucht. Des Weiteren wurde die Wichtigkeit der Echtzeitsimulation für detaillierte Fahrzeugmodelle verdeutlicht. Dabei bestand das Ziel darin, ein echtzeitfähiges Simulationsmodell zu entwickeln, dass die Fahr- und Antriebsstrangdynamik möglichst genau abbildet. Außerdem sollte das Fahrzeugmodell in der ganzen Entwicklungsphase von Fahrzeugregelungen einsetzbar sein.

Auf der Grundlage der Bibliotheken „VehicleDynamics“ und „PowerTrain“ wurden mit *Dymola* objektorientierte Fahrzeugmodelle erstellt. Für den Antriebsstrang wurde ein Modell erstellt, dass die Strukturumschaltungen in den Schaltelementen des Getriebes berücksichtigt. Darüber hinaus wurde eine Getriebesteuerung mit Beeinflussung des Motormoments für die Steuerung der Gangwechsel erstellt. Die Struktur des erstellten, nichtlinearen, dynamischen Modells beruht auf den Komponenten realer Fahrzeuge. Für die Simulationsun-

tersuchungen wurde ein Fahrermodell erstellt, dass die Fahrzeuggeschwindigkeit nach einem vorgegebenen Fahrzyklus regelt und einen Lenkwinkelverlauf vorgibt.

Die verwendeten Simulationssysteme *Dymola* und *xPC Target* ermöglichen eine kostengünstige und effiziente Modellierung sowie Offline- und Echtzeitsimulation von *Modelica*-Modellen. *Dymola* ermöglicht mit der grafischen Oberfläche eine schnelle Modellierung, Parametrisierung und Initialisierung des Fahrzeugmodells. Bei der Modellerstellung ist jedoch auf eine vorteilhafte Festlegung der Modellparameter und deren Struktur zu achten. Die von *Dymola* erstellte Echtzeitanwendung kann einfach mittels *xPC Target* und dem *Realtime-Workshop* von *The MathWorks* auf einem Arbeitsplatzrechner simuliert werden.

Das detaillierte Fahrzeugmodell stellt alle notwendigen Größen zum Test und Optimierung von Fahrdynamik- und Antriebsstrangsteuergeräten zur Verfügung. Die erstellten Modelle wurden verifiziert und wesentliche Ergebnisse dargestellt. Das Fahrzeugmodell muss jedoch für Antriebs- und Fahrdynamikuntersuchungen oder zum Testen und Optimieren von Steuergeräten entsprechend Parametrisiert werden.

*Modelica* eignet sich, mit den Simulationssystemen *Dymola* und *xPC Target*, hervorragend um Gesamtfahrzeugmodelle zu erstellen und in Echtzeit zu simulieren. Damit wird der komplette Entwicklungsprozess für komplexe Steuerungs- und Regelungsfunktionen wie beispielsweise das ESP oder das Antriebsstrangmanagement unterstützt. Es wurde gezeigt wie ein objektorientiertes Gesamtfahrzeugmodell zu erstellen ist und die Echtzeitsimulation mit *xPC Target* erfolgt. Durch den effizienten Echtzeitcode ist es außerdem möglich, das Fahrzeugmodell Online auf einem entsprechenden Rechensystem im Fahrzeug einzusetzen. Es bestehen Bibliotheken mit denen die 3D-Effekte im Antriebsstrang auf die Fahrzeugdynamik untersucht werden können. Des Weiteren eignet sich das System auch um detaillierte Teilkomponenten zu simulieren und zu testen. Besonders im Bereich der Mechatronik sollte *Modelica* durch den multidisziplinären Ansatz besondere Vorteile bei der Modellierung und Simulation bieten. Durch den Fortschritt bei der Rechnerentwicklung sind in absehbarer Zeit weitere Detaillierungen des Fahrzeugmodells möglich.

# A Matlab-Skript

Nachfolgend ist das Skript zur Speicherung der Ergebnisse einer Echtzeitsimulation vom „Target-PC“ in den Workspace von *Matlab* dargestellt:

```
%Variablen in den Workspace kopieren
X=tg.Outputlog;
time=tg.TimeLog;
TET=tg.TETLog;
AvgTET=tg.AvgTET;
MinTET=tg.MinTET;
MaxTET=tg.MaxTET;

%X-Variable nach den einzellenen Ausgangsvariablen aufteilen
desiredSpeed=X(:,1);
throttle=X(:,2);
brake=X(:,3);
selectedGear=X(:,4);
engineSpeed=X(:,5);
gearOutputSpeed=X(:,6);
r0F=X(:,7:9);
F_w=X(:,10:12);
F_z=X(:,13:15);
F_v=X(:,16:18);
F_a=X(:,19:21);
clutchPressure_C=X(:,23);
clutchPressure_D=X(:,22);
clutchMode_C=X(:,24);
clutchMode_D=X(:,25);
M_E=X(:,26);
M_T=X(:,27);
M_W3=X(:,28);
M_W4=X(:,29);

clear X;
```

## B Dymola-Skript

Nachfolgend ist das *Dymola*-Skript für die Erstellung von schnellen Echtzeitcode dargestellt:

```
Evaluate:=true; //Erhöhung der Rechenperformance

//Verwenden des Dymola-Interpreters
//mit dem Intergrationverfahren Explizit Euler
//und einer Schrittweite von 0.001s
Advanced.InlineMethod:=1;
Advanced.InlineFixedStep:=0.001;

// Hilft dem Compiler schneller Code zu erstellen
Hidden.InlineIntegrationJacobianForVariableStep = false;
Hidden.InlineIntegrationSeparateJacobian=true;

// Einstellungen für eine schnellere Echtzeitsimulation
Hidden.Define.NewJacobian=false;
Hidden.Define.AimForHighAccuracy=false;
Hidden.Define.AutoRemoveAuxiliaries=true;
Hidden.InterpolateStatesForInline=true;
```

# Literaturverzeichnis

- [Bee01] BEER, Alfred: *X-by-Wire: Von der Entwicklung zur Einführung*. ATZ – Automobiltechnische Zeitschrift Sonderheft, 2001
- [Beu00] BEUSCHEL, Michael: *Neuronale Netze zur Diagnose und Tilgung von Drehmomentschwingungen am Verbrennungsmotor*. Dissertation, TU München Fakultät für Elektrotechnik und Informationstechnik, 2000
- [BJT<sup>+</sup>01] BORGMANN, Klaus ; JÄGERBAUER, Ernst ; TISCHER, Jens ; HALL, Wolfgang ; BOCK, Christian ; SCHOPP, Johann: *Der Antrieb des neuen BMW 7er*. ATZ – Automobiltechnische Zeitschrift Sonderheft BMW 7er, 2001
- [BOS02] BOSCH: *Kraftfahrtechnisches Taschenbuch*. 24. Auflage, Vieweg Verlag, 2002
- [Cim03] CIMANDER, Gerald: *Echtzeitsimulation und Optimierung eines Automatikgetriebes*. Diplomarbeit, TU München Fakultät für Maschinenwesen, 2003
- [Cra99] CRAMPEN, Heinz Manfred: *Fahrautomat für Pkw-Antriebsstränge*. Dissertation, Fakultät für Maschinenwesen der Technischen Hochschule Aachen, 1999
- [DGK01] DACH, Hansjörg ; GRUHLE, Wolf-Dieter ; KÖPF, Peter: *PKW-Automatgetriebe*. Landsberg: Verlag Moderne Industrie, 2001
- [Gau00] GAUPP, Winfried: *Elektronik in Bremssystemen*. ATZ – Automobiltechnische Zeitschrift 102 (2000) 2, 2000
- [Güh03] GÜHMANN, Clemens: *Einsatz der Simulation in der Applikation und im automatisierten Test von Getriebesteuerungen*. IIR Tagung Getriebelektronik, Regensburg, Juni 2003
- [Haj02] HAJ FRAJ, Ali: *Dynamik und Regelung von Automatikgetrieben*. Dissertation, VDI Verlag, Fortschr.-Ber. VDI Reihe 12 Nr. 489, 2002

- [Hel03] HELLER, Stefan: *Fahrdynamikmodellierung, Reglerimplementierung und Simulation mit MODELICA/DYMOLA*. Diplomarbeit, TU München Lehrstuhl für Realtime-Computersysteme, 2003
- [Hol01] HOLZMANN, Henning: *Adaptive Kraftfahrzeugdynamik-Echtzeitsimulation mit Hybriden Modellen*. Dissertation, VDI Verlag, Fortschr.-Ber. VDI Reihe 12 Nr. 465, 2001
- [Ise03] ISERMANN, Rolf: *Steuerung und Regelung im Auto*. ATZ – Automobiltechnische Zeitschrift 51 (2003) 7, 2003
- [JVWW01] JELDEN, Hanno ; VELDTEN, Burkhard ; WEISS, Norbert ; WOLGAST, Bernd: *Elektronische Steuerung von Dieselmotoren*. ATZ – Automobiltechnische Zeitschrift, Sonderausgabe, 2001
- [KG00] KESSLER, Frank ; GEBERT, Jürgen: *Testautomatisierung und Antriebsstrangmodellierung an HIL-Steuergeräteprüfständen in der BMW Getriebeentwicklung*. ATZ – Automobiltechnische Zeitschrift 102 (2000) 5, 2000
- [KN98] KRAMER, Ulrich ; NECULAU, Mihaela: *Simulationstechnik*. Carl Hanser Verlag, 1998
- [Koc01] KOCH, Jochen: *Modellbildung und Simulation eines Automatikgetriebes zur Optimierung des dynamischen Schaltungsablaufs*. Dissertation, Fakultät Physik der Universität Stuttgart, 2001
- [Lam01] LAMBERG, Klaus: *Methodik zur systematischen Bereitstellung von HIL-Testsystemen für Kfz-Steuergeräte*. Herbert Utz Verlag, 2001
- [Las88] LASCHET, Andreas: *Simulation von Antriebssystemen*. Springer Verlag, 1988
- [LN94] LECHNER, Gisbert ; NAUNHEIMER, Harald: *Fahrzeuggetriebe*. Springer-Verlag, 1994
- [Mau03] MAUERMANN, Dirk: *Modellierung und Simulation von Otto-Motoren im Kraftfahrzeug*. Belegarbeit, HTWK Leipzig Institut für Mess-, Steuerungs- und Regelungstechnik, 2003
- [Met03] METTIN, Uwe: *Entwicklung eines linearen Einspurmodells zur Untersuchung der Querdynamik zweiachsiger Kraftfahrzeuge*. Forschungsbeleg, HTWK Leipzig Institut für Mess-, Steuerungs- und Regelungstechnik, 2003



- [Mit90] MITSCHKE, Manfred: *Dynamik der Kraftfahrzeuge - Band C*. Springer-Verlag, 1990
- [Mit95] MITSCHKE, Manfred: *Dynamik der Kraftfahrzeuge - Band A*. Springer-Verlag, 1995
- [MODa] *MODELICA Libraries*. URL <http://www.modelica.org/libraries.shtml>
- [MODb] *Open Modelica*. URL <http://www.ida.liu.se/~pe-lab/modelica/OpenModelica.html>
- [PS93] POPP, Karl ; SCHIEHLEN, Werner: *Fahrzeugdynamik*. Stuttgart : Teubner, 1993
- [PSO66] PELCHEN, Christoph ; SCHWEIGER, Christian ; OTTER, Martin: *Modeling and Simulating the Efficiency of Gearboxes and of Planetary Gearboxes*. MODELICA'02, S. 257-266
- [Ril94] RILL, Georg: *Simulation von Kraftfahrzeugen*. Friedr. Vieweg und Sohn Verlagsgesellschaft mbH, 1994
- [RRS03] RICHERT, Felix ; RÜCKERT, Joachim ; SCHLOSSER, Axel: *Vergleich von Modelica und Matlab anhand der Modellbildung eines Dieselmotors*. at – Automatisierungstechnik 51, 2003
- [Run01] RUNGE, Wolfgang: *Elektronikentwicklung bei Automatikgetriebe und Lenkung*. ATZ – Automobiltechnische Zeitschrift, Sonderausgabe, 2001
- [SBT<sup>+</sup>03] SCHWARZ, Ralf ; BAUER, Urs ; TRÖSTER, Steffen ; FRITZ, Stefan ; MUNTU, Matthias ; SCHRÄBLER, Sighard ; WEINREUTER, Markus ; MAURISCHAT, Christian: *ESP II Fahrdynamik der nächsten Generation*. ATZ – Automobiltechnische Zeitschrift 105 (2003) 11, 2003
- [Sch99] SCHERNEWSKI, Ralf: *Modellbasierte Regelung ausgewählter Antriebssystemkomponenten im Kraftfahrzeug*. Dissertation, Fakultät für Elektrotechnik der Universität Fridericiana zu Karlsruhe, 1999
- [SOC04] SCHWEIGER, Christian ; OTTER, Martin ; CIMANDER, Gerald: *Objektorientierte Modellierung mit Modelica zur Echtzeitsimulation und Optimierung von Antriebssträngen*. AUTOREG, VDI-Verlag, (VDI-Berichte 1828), pp. 639-650, 2004
- [Som00] SOMMER, Steffan: *Verknüpfung von horizontaler und vertikaler Dynamik*. ATZ – Automobiltechnische Zeitschrift, Sonderausgabe, 2000

- [SR03] SCHWARZ, Ralf ; RIETH, Peter: *Global Chassis Control - Systemvernetzung im Fahrwerk*. ATZ – Automobiltechnische Zeitschrift 51 (2003) 7, 2003
- [SRG03] SCHMIDT, Karsten ; RIESE, Jens ; GÜHMANN, Clemens: *Testen – Integraler Bestandteil der Entwicklung?* 1. Tagung Simulation und Test in der Funktions- und Softwareentwicklung für die Automobilelektronik, Berlin, Oktober 2003
- [SZ03] SCHÄUFFELE, Jörg ; ZURAWKA, Thomas: *Automotive Software-Engineering*. Wiesbaden : Vieweg, 2003
- [TEV] *Continental Teves*. URL <http://www.contiteves.de>
- [Tsc02] TSCHARNUTER, Dietmar: *Optimale Auslegung des Antriebsstrangs von Kraftfahrzeugen*. Dissertation, TU München Fakultät für Mathematik, 2002
- [vS02] VAN BASSHYSEN, R ; SCHÄFER, F: *Handbuch Verbrennungsmotor*. 2. verbesserte Auflage, Vieweg Verlag, 2002
- [Wil98] WILLUMEIT, Hans-Peter: *Modelle und Modellierungsverfahren in der Fahrzeugdynamik*. B. G. Teubner Stuttgart • Leipzig, 1998
- [WN03] WIEDEMANN, Jochen ; NEUBECK, Jens: *Fahrdynamikentwicklung*. ATZ – Automobiltechnische Zeitschrift 105 (2003) 12, 2003

## **Eidesstattliche Erklärung**

Hiermit erkläre ich an Eides Statt, dass ich die vorliegende Studienarbeit selbständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht habe, was aus Arbeiten anderer unverändert oder mit Änderungen entnommen wurde.

Leipzig, den 14.Juli 2004